AN EXEMPLAR-BASED APPROACH TO SEARCH-ASSISTED COMPUTER-AIDED DIAGNOSIS OF PIGMENTED SKIN LESIONS

A Dissertation
Presented to
The Academic Faculty

by

Zhen Hao (Howard) Zhou

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing,
College of Computing

Georgia Institute of Technology
December 2010

AN EXEMPLAR-BASED APPROACH TO SEARCH-ASSISTED COMPUTER-AIDED DIAGNOSIS OF PIGMENTED SKIN LESIONS

Approved by:

Professor James M. Rehg,
Committee Chair
School of Interactive Computing,
College of Computing
Georgia Institute of Technology

Professor Irfan Essa
School of Interactive Computing,
College of Computing
Georgia Institute of Technology

Dr. Mei Chen
Intel Labs Pittsburgh
Intel Corp.

Professor Thad Starner
School of Interactive Computing,
College of Computing
Georgia Institute of Technology

Professor Greg Turk
School of Interactive Computing,
College of Computing
Georgia Institute of Technology

Date Approved: November 12, 2010

This thesis is dedicated to my parents,

who made everything possible for me.

ACKNOWLEDGEMENTS

I am grateful to my advisor, Dr. James M. Rehg, and my committee members Dr. Mei Chen, Dr. Greg Turk, Dr. Irfan Essa, and Dr. Thad Starner, whose encouragement, supervision and support from the preliminary to the concluding level enabled me to develop an understanding of the subject. I am also grateful to Dr. Jay Summit and Dr. Tracy Westeyn for their constructive and thoughtful suggestions on the design and execution of the user study, and to Dr. Laura K. Ferris and Dr. Alexander Zhang for their instructional comments on general dermatology related topics, and to Dr. Richard Gass and Dr. Aaron Cois for their ingenious design of the DermFind image acquisition device, which enabled the subsequent development of the interactive user interface. Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of this disseration.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Over the years, exemplar-based methods have yielded significant improvements over their model-based counterparts in image synthesis applications. Notably, texture synthesis algorithms using an exemplar-based approach have shown success where traditional stochastic methods failed. As an illustrative example, I present an exemplar-based approach that yields substantial benefits for user-guided terrain synthesis using Digital Elevation Models (DEMs). This success is realized through exploitation of structural properties of natural terrain. Recently, the development of algorithms and data structures that facilitate search in high-dimensional space has led to a proliferation of exemplar-based methods that synthesize image contents and style by tapping into a large collection of images. In addition, as annotated image datasets become increasingly available, the exemplar-based approach is gaining in popularity for image analysis applications.

This thesis addresses the intersection between exemplar-based analysis and the problem of content-based image retrieval (CBIR). A basic problem in CBIR is the process by which the search criteria are refined by the user through the manipulation of returned exemplars. Exemplar-based analysis is particularly well-suited to query refinement due to its interpretability and the ease with which it can be incorporated into an interactive system. I investigate this connection in the domain of Computer-Assisted Diagnosis (CAD) of dermatological images. I will use the analysis of dermatological images using CBIR as context to demonstrate that exemplar-based methods can also yield significant benefits for image analysis if analogous structural properties can be identified. I will present an exemplar-based algorithm for segmenting pigmented skin lesions in dermoscopy images. In addition, I will present a generalized

representation of dermoscopic feature for detection and matching. This representation enables us to realize interactive region of interest (ROI) retrieval capability, including a relevance feedback mechanism to facilitate more flexible query-by-example analysis. Finally, I will assess the role of the benefit of this CBIR-CAD approach through both quantitative evaluations and user studies.

CHAPTER I

INTRODUCTION

Exemplar-based methods can be characterized by having the following four elements, 1) a set of exemplars, 2) a target about which we would like to make a prediction, 3) a mechanism to identify neighbors of the target based on some relevance measure, and 4) a mechanism to transfer information from the neighbors to the target, resulting in a predicted output. The exemplar-based approach essentially invokes a transduction process, reasoning from specific observed cases to specific unobserved cases. In contrast to the model-based approach, it avoids the need to solve a more general problem - inferring general rules from observed cases, before solving a more specific problem - computing outputs for new cases.

1.1 Success of the exemplar-based approach in the image synthesis domain

Over the years, exemplar-based methods have yielded significant improvement over their model-based counterparts in the image synthesis domain. Notably, texture synthesis algorithms using an exemplar-based approach have shown success where traditional stochastic methods failed. Image-based texture synthesis is the process of creating an arbitrarily large patch of texture by drawing pixels from a given example image. Over the past decade, a steady improvement in the quality of synthesized textures, both 2D and volumetric, has been achieved through an evolution from pixel-based methods [46] to non-parametric neighborhood-based methods [29, 100, 3]. The most recent patch-based techniques, exemplified by [28, 47, 58, 103, 7, 59, 63, 64] have achieved the most visually-compelling results. In the same spirit, Hertzmann et al. have introduced an image analogy framework that is capable of transferring

various artistic styles of the content from one image to another [47].

These texture synthesis algorithms use a single example image as the source. Building on this approach, many algorithms use an analogous approach to perform synthesis using a collection of images. Freeman et al. [37] proposed an image interpolation algorithms using a database of training images to create plausible high-frequency details for image super-resolution. Recent work [26, 101, 54, 44, 60] has shown that novel image contents can be synthesized by manipulating a collection of images. Wilczkowiak *et al.* [101] have demonstrated that for the task of object removal, searching through multiple images to collect potentially useful image regions can significantly improve the result, especially when the contents to be filled in are structured, man-made environments. Diakopoulos *et al.* [26] and Johnson *et al.* [54] have shown that a wider range of semantically valid image contents can be synthesized provided with an image database with labeled regions. The labeling process is supervised in [26] and semi-supervised in [54]. In addition, Hays and Efros [44] have shown that the good results can be achieved even without annotations or labeling by the users. Their algorithm fills in holes by finding similar image regions from the database. The synthesized images are not only seamless but also semantically valid. To address the issue that a low-level scene descriptor in and of itself cannot encode high-level semantic information, they observed that scenes can only be trusted to be semantically similar if the distance between them is very small, so their solution is to collect a huge number (two million) of images so that it is more likely that a very similar scene is available in the dataset. They reported that a qualitative leap in performance occurs when the number of images increased from ten thousand to two millions. The photo clip art work of [60] essentially uses a similar approach to find semantically valid objects in a large dataset and insert into the background to create a new image. To take this one step further, Liu et al. have demonstrated synthesizing motion via transfer of moving objects using the SIFT flow algorithm. The central

step in the SIFT flow algorithm is to align video frames to their neighbors in a large image collection consisting of a variety of scenes [67]

The success of the exemplar-based approach over the model-based generative approach in the context of image synthesis arises from its transduction property, i.e. it avoids the need to solve a more general problem (inferring general rules from observed cases) before solving a more specific problem (computing outputs for new cases). For instance, it has proven to be very difficult to control the capacity of a standard generative texture model so as to balance the twin concerns of being general enough to represent the many forms of texture that exist in the world, and being restrictive enough to avoid over-fitting to a limited training set. Similar issues arise for the task of synthesizing semantically valid image content. In contrast, algorithms that copy their outputs from parts of existing images are more likely to preserve semantic validity. The task is then reduced to finding plausible regions that are semantically compatible with the target and merging them seamlessly.

1.2 Using the exemplar-based approach in image analysis

Over the years, exemplar-based image synthesis methods have received significant attention. In contrast, model-based methods are still the predominant force in image analysis, in part because of the great success of modern methods such as SVM and AdaBoost. This is not to say that the exemplar-based approach has no merit. On the contrary, exemplar-based methods are conceptually simple to understand because they depend on simple locality arguments and are easy to implement using clustering and interpolation algorithms. Moreover, the ability to examine the returned exemplars in an analysis task gives exemplar-based methods an explanatory property that are unmatched by black-box style model-based methods.

However, there have been some major obstacles for solving image analysis problems using the exemplar-based approach. First of all, for problems with high dimensional features, even with an extremely large training set, the points can be relatively sparse, making it difficult to find a close match. This is not a significant problem in the context of image synthesis since a large number of examples can be acquired inexpensively and the pixel values alone are sufficient. However, inference problems require that a large number of annotations must be provided along with the exemplars, which requires the investment of significant manual labor. Moreover, algorithms and data structures for identifying neighbors often reduce to linear search when dealing with high dimensional features, rendering exemplar-based methods much less efficient than model-based approaches. Fortunately, recent developments in related areas of computer science have alleviated some of these problems and have made exemplar-based analysis methods much more feasible. On the one hand, large annotated image datasets such as LabelMe [85], TinyImages [94], and LHI [104], etc, are becoming increasingly available. On the other hand, the development of data structures that facilitate search in a high-dimensional metric space, e.g., various metric trees [105] and Locality Sensitive Hashing (*LSH*) methods [40], etc, make searching large databases efficient enough for practical purposes. As a result, researchers have made considerable progress in developing exemplar-based methods for image analysis tasks. Hayes and Efros have proposed an algorithm using a data-driven scene matching approach for estimating a distribution over geographic locations from a single image [45]. Malisiewicz and Efros developed an exemplar-based object recognition framework where a predefined image segment represents an instance in a particular object class. By learning per-exemplar distances, they have demonstrated the utility of this recognition by association approach in image understanding tasks. Divvala et al. investigated the possibility of utilizing large amounts of unlabeled image data to help in the estimation of rough scene layout [87]. Liu et al. have proposed to use

the SIFT flow algorithm to predict motion field from a single static image [67]. In an effort to address the search efficiency issue when a large number of high quality labeled images are provided, Torralba et al. have showed that using compact binary code on the LabelMe [85] dataset gives surprisingly powerful recognition results using simple nearest neighbor techniques [95].

1.3 Addressing the need to support interpretability and interactivity

One of the properties that makes the exemplar-based approach attractive is the availability of exemplars as an explanatory means. Consequently, this makes the exemplar-based analysis framework particularly well-suited for content-based image retrieval (CBIR) for computer aided diagnosis (CAD). For instance, a doctor using a particular CAD system may want to understand why a certain diagnosis recommendation was made by the system. If the inference rule used is exemplar-based, she can browse through the neighbors and obtain an intuitive understanding of the explanation for the diagnosis. In contrast, it is much more challenging to obtain insight for black-box model-based inference methods.[1] In addition, if she is not satisfied with the diagnosis and has a basic understand of some undesirable or irrelevant properties of the returned exemplars, she can provide her feedback through manipulation of the exemplar set. In contrast, if model-based inference method was used to produce such diagnosis, to provide useful feedback would require the doctor to have understanding over some obscure system specific parameters. *The need for interpretability and interactivity in CAD makes the exemplar-based approach an especially attractive choice.*

[1] In the case of linear SVM's, for example, it is sometimes possible to visualize the weights that are associated with feature elements (see Fig.6 in [25]). However, this visualization is usually not feasible for more complex (and powerful) kernels.

1.4 Thesis statement

The exemplar-based approach to CBIR for CAD enables ROI retrieval and relevance feedback mechanism which result in improved diagnoses for pigmented skin lesions.

1.5 Contributions

This work focuses on exploring exemplar-based approaches in the domain of CBIR-CAD for PSL diagnosis, we summarize our contributions as follows: In the domain of skin cancer diagnosis,

- we have designed and implemented the first end-to-end CBIR-CAD system, which houses one of the largest annotated dermoscopy image repository, and
- we are the first to quantitatively assess the utility of such CBIR-CAD systems through both component-wise evaluations and a user-study involving real experts in the field.

In addition, we have created two new CBIR-CAD capabilities for skin cancer diagnosis,

- we have designed various exemplar-based methods for many algorithmic components of the DermFind system, including artifact removal, segmentation, and diagnosis, and
- we are the first to provide interactive Region-of-Interest (ROI) retrieval and relevance feedback mechanism to facilitate user interaction in order to improve CBIR-CAD performance in the domain of skin cancer diagnosis.

1.6 Organization

The dissertation document is organized as follows. Chapter 2 describes a novel application of exemplar-based synthesis to the problem of terrain synthesis using Digital

Elevation Models. Chapter 3 gives an overview of DermFind, a CBIR based CAD system for dermatological cases, which will be used as the context for developing various exemplar-based image analysis techniques. Chapter 4 describes the first of such techniques, a artifact removal procedure that preserves lesion features via exemplar-based inpainting. Chapter 5 focuses on an exemplar-based segmentation method for pigmented skin lesions in dermoscopy images. Chapter 6 lays out a generalized representation of dermoscopic feature for detection and matching. This representation enables the interactive image and region-of-interest search, which will be discussed in detail in chapter 7. Finally, chapter 8 concludes with experiments and user studies for the evaluation the DermFind system.

CHAPTER II

EXEMPLAR-BASED TERRAIN SYNTHESIS FROM DIGITAL ELEVATION MODELS

In this chapter we describe a novel application of exemplar-based synthesis to the problem of terrain synthesis using Digital Elevation Models (*DEMs*). This work was inspired by the success of exemplar-based approaches in the texture synthesis domain. Our method uses high-resolution DEMs as exemplars to synthesize realistic-appearing terrain. The exemplar-based nature of this method also enables a sketch-based specification of synthesized terrain features, giving the user intuitive control over the synthesis result.[1]

2.1 Introduction

There are numerous applications that make use of synthetic terrain, and very often the terrain is the dominant visual element in the scene. Such applications include landscape design, flight simulators, emergency response training, battleground simulations, feature film special effects and computer games. Over the years, graphics researchers have made considerable progress towards developing efficient methods for generating synthetic terrain. Previous terrain synthesis work has focused on using fractal models and physical erosion models to create realistic-appearing terrain.

With the rapid growth of computing power and development in terrain visualization techniques, the demand for more realistic terrain has increased considerably. In addition, users of terrain modeling applications want more control over the creation of new terrain. However, current terrain synthesis methods have several limitations.

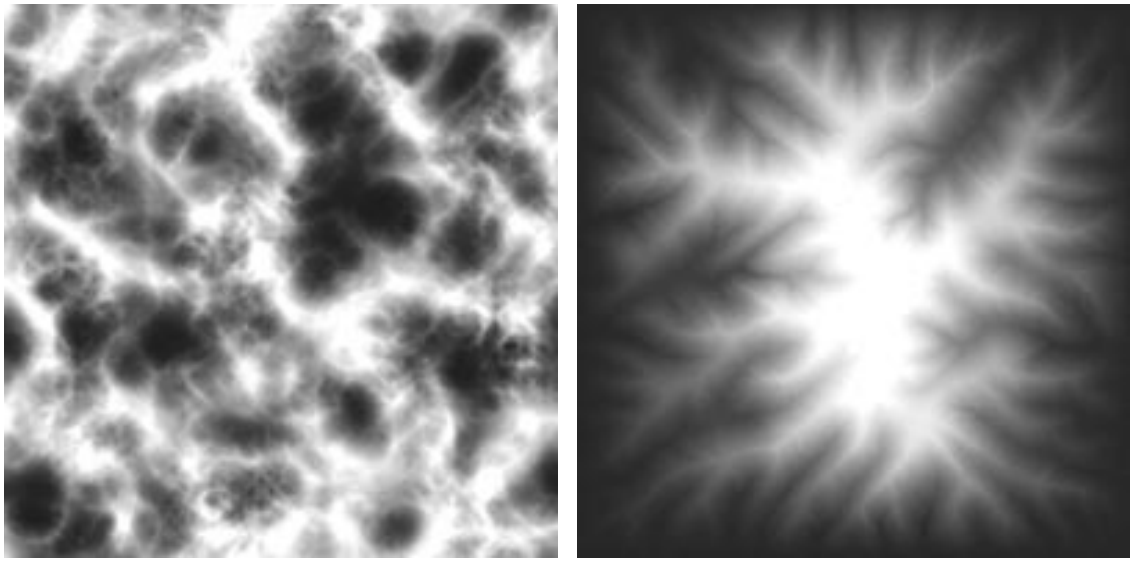
[1]The work described in this chapter has been published in [111].

First, these methods provide users with little or no control over the placement of desired terrain features. Second, using the control parameters in these methods, it is difficult to generate terrain with a desired style, such as a terrain with the geological features of the Grand Canyon.

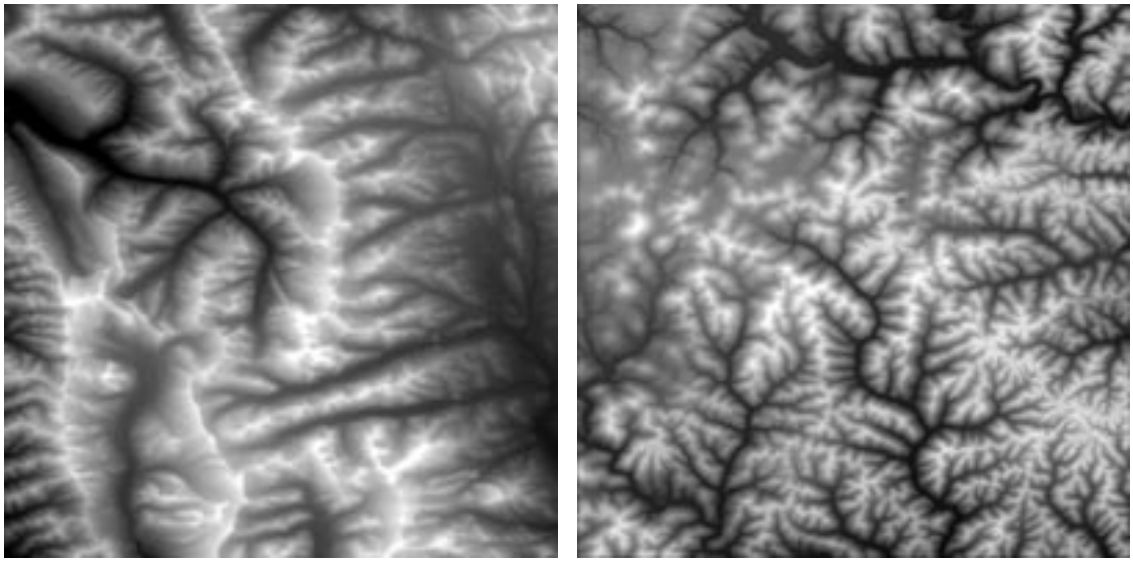
Figure 1: Synthesized terrain using DEM of Flathead National Forest Mountain Range (top image) and DEM of the Grand Canyon (bottom image).

Fig. 2(a) and 2(b) show examples of fractal and erosion terrain produced by some popular commercial software, displayed as intensity-coded elevation maps. The styles of these terrains are quite unlike the natural terrain illustrated in Fig. 2(c) and 2(d).

We present a novel example-based terrain synthesis method which addresses the need for intuitive user control over both terrain feature placement and terrain style. Our method draws upon the techniques of patch matching and patch placement from



(a) (b)



(c) (d)

Figure 2: Examples of terrain generated by current methods. (a) Fractal Terrain (Ridged Multi Perlin) generated by Terragen at 1025×1025 resolution. (b) Erosion Terrain generated by Bryce 5 at 1024×1024 resolution. In contrast are two real elevation maps: (c) 1/3 arc second(10m) DEM of Flathead national forest mountain range, MT and (d) 1/3 arc second(10m) DEM of Mount Vernon, KY. (DEM courtesy of the U.S. Geological Survey)

example-based texture synthesis. In our approach, the user supplies a sketched terrain feature map (called the *sketch map*) and real terrain data (called the *example height field*) which contain desired terrain styles. Example height fields are in the form of Digital Elevation Models (DEMs), which are available online from the U.S. Geological Survey. Our system then automatically generates a new height field that preserves the visual style of the real terrain data and meets the feature constraints of the sketch map. Synthetic results from our approach are shown in Fig. 1.

The sketch map provides the user with an easy and intuitive way to control the synthesis process. Each map specifies the locations of important terrain features, such as the bifurcation point at the center of Fig. 10(a). Notice that these sketches are quite coarse. In fact, the width of the brush and the pixel intensities are of little importance, as long as they follow the simple principle that darker indicates lower elevation and brighter indicates higher elevation. Our goal in this work is the generation of visually-compelling terrain. We do not address the separate issue of whether a synthesized terrain is geologically accurate.

The starting point for our algorithm is the identification of important terrain features in both the sketch map and the example height field. Our system concentrates on large curvilinear features such as rivers, valleys and mountain ridges, since these are usually the most important visual elements in large-scale terrain (as illustrated in Fig. 2(c) and 2(d)). Because the underlying terrain structures in the sketch map and the ones in the example height field are often very different, matches over large spatial scales are unlikely to be correct. However, we observe that at small scales, common terrain features such as a bifurcation or a straight section of valley can be found in both maps. Hence, our algorithm breaks the sketch map into small patch regions and searches through the example height field for structural feature matches.

Extraction of structural information from the sketch map is straightforward due to its simplicity. On the other hand, the extraction of high-level structural information

such as ridges and valleys from a real terrain height field can be difficult. Here we draw upon work in the field of geomorphology. Specifically, we use the Profile recognition and Polygon breaking Algorithm (*PPA*) developed by Chang et al. [18] to extract structural information such as ridge and valley axes from a height field.

Once the important structural terrain features have been extracted, our algorithm proceeds in the following stages. First, curvilinear terrain features are used as constraints for the matching and alignment of patch regions in the sketch map and the example height field. Second, these features are used as constraints for matching along the overlapping boundary between neighboring patch regions. A patch is selected based on a weighted combination of scores which assess the goodness-of-fit between a patch and its neighbors and the amount of deformation of the patch that is required. The order of patch placement is determined by a breadth-first traversal of a feature tree which is constructed from the user sketch. Each patch is placed into the output map using a combination of graph cut seam finding and Poisson seam removal to minimize visual discontinuities.

In this work we make the following contributions:

- Method for sketch-based specification of synthesized terrain features, giving the user intuitive control over the synthesis result.
- Feature-based approach to matching and placement of large curvilinear terrain features, which makes it possible to efficiently search large terrain databases and preserve important visual elements in the synthesis process.
- Tree-ordered patch placement algorithm, as an alternative to standard raster-scan ordering, which results in a more faithful reproduction of terrain structure in matching the user input.
- The ability to synthesize transitions between different terrain types and incorporate multiple DEMs into a single synthesized output.

We have drawn significant inspiration for our work from previous texture synthesis methods, which we review in the following section.

2.2 Related work

There are two main approaches to generating synthetic terrain: fractal landscape modeling and physical erosion simulation. Fractal landscape modeling dates back to the pioneering work of Mandelbrot [69]. Since then, a variety of stochastic subdivision techniques have been introduced. Fournier et al. [36] introduced the random midpoint displacement technique to create fractal surfaces. Voss [99] added successive random displacement to fractional Brownian surfaces. Miller [73] proposed a square-square subdivision scheme for generating fractal terrain and a parallel processing algorithm for rendering height fields. Lewis [65] proposed generalized stochastic subdivision. Szeliski and Terzopoulos [92] addressed the problem of user control by combining deterministic splines and stochastic fractals into constrained fractals. Recent fractal-based approaches are reviewed in [27] and [24].

Physical erosion simulation is an alternative approach to synthesizing terrain details based on models of landscape formulation and stream erosion from the geomorphology community. It is often used as a refinement step after a rough height field is generated. Kelley et al. [55] first introduced a method to approximate natural terrain by simulating the erosion of stream networks. Later, Musgrave et al. [74] then combined the fractal modeling and erosion simulation approaches into a single framework. Recent physical erosion techniques, exemplified by [83], [19], [75], [5], and [76], have focused on improving both the physical modeling aspect and computational efficiency. With appropriately-tuned parameters, these techniques can generate realistic-appearing terrain.

Both fractal and physical erosion techniques add terrain details through procedural refinement, which often involves non-intuitive parameter tuning. Recently, Brosz et

al. [11] attempt to extract high resolution terrain details from existing DEM data and apply it to lower resolution terrain through multi-resolution analysis. In practice, their method requires both the source and target terrain to be fairly detailed and does not grant the user freedom to create arbitrary terrain.

To provide the user with more intuitive control over the synthesized terrain, image-based alternatives were proposed by Lewis [66] and Perlin and Velho [78]. In these works, terrain was viewed as a type of texture and user control was provided through direct manipulation of the texture. The result was then interpreted as a height field to create a variety of terrain types. However, because it is difficult for a user to draw a realistic natural height field by hand, these methods typically suffer from a lack of realistic detail.

Commercial Software We are not aware of any existing commercial software for terrain synthesis that employs an approach that is similar to ours. We briefly review four major commercial systems, Mojoworld, Terragen, World Machine, and Bryce, which are representative of the current state-of-the-art. Mojoworld and Terragen both have fractal synthesis engines which generate terrain procedurally. The resulting models are very compact because they are generated on-the-fly. World Machine, on the other hand, offers geological erosion on top of its procedural shape and noise generator to synthesize realistic-appealing terrain. In all these systems, control over the synthesis result is obtained by changing global parameters in the generation process. It is, however, difficult to set these parameters so as to generate distinctive, realistic terrain types. Moreover, these systems do not support the user-specified placement of major terrain features. Bryce, in addition to having both fractal and erosion synthesis engine, also accepts as input a direct specification of the height field by the user using a painting approach (essentially the method of [66] with the addition of fractal noise). Fig. 2(c) and 2(d) are example terrains generated by such commercial systems. None of these products, however, make it possible to synthesize

terrain in a specific style, such as style of the Grand Canyon.

Texture Synthesis Image-based texture synthesis is the process of creating an arbitrarily large patch of texture by drawing pixels from a given example image. During the past few decades, a steady improvement in the quality of synthesized textures, both 2D and volumetric, has been achieved through an evolution from pixel-based methods [46] to non-parametric neighborhood-based methods [29], [100], [3]. The most recent patch-based techniques, exemplified by [28], [47], [58], [103], [7], [59], [63], and [64], have two common stages: 1) search in a sample texture for neighborhoods most similar to a context region; 2) merge a patch or a pixel with the (partially) synthesized output texture. Dynamic programming [28] and graph cuts [58] have been used to optimize the patch merging stage. We employ a related search-and-merge strategy that addresses the unique characteristics of terrain data.

Recently, Zhang et al. [106] introduced feature-based warping and blending techniques to synthesize progressively-variant texture on arbitrary surfaces. In their work the feature texton masks were manually extracted. Wu and Yu [103] use edges extracted from the input texture as high-level features to guide patch-based texture synthesis. We build on this earlier work in two ways. First, we employ curvilinear features to support user-sketching of desired terrain features and efficient search for matching patches in large terrain datasets. Second, we introduce a feature analysis technique which can reliably extract global terrain characteristics such as ridges and valleys from large DEMs in a wide range of styles. We demonstrate that standard edge-finding methods are inappropriate for this task.

The image analogies framework introduced by Hertzmann et al. [47] can be used to synthesize terrain images through a texture-by-numbers approach. This work does not directly synthesize novel terrain height fields. Its application to height field synthesis is hampered by the difficulty of guaranteeing that the local neighborhood matching approach would preserve extended structures such as ridges and valleys that

characterize terrain style. The GPU-based texture synthesis method presented in [63] includes drag-and-drop features and synthesis magnification. These techniques were used to relocate mountains on a terrain height map. In contrast to this work, our focus is to automatically generate a complete terrain image in a particular style by matching extended terrain features with a desired user sketch. We therefore support fully-automatic feature extraction and matching.

While our example-based approach to terrain synthesis is inspired by the recent success of patch-based methods, terrain synthesis is not simply texture synthesis on height fields. The terrain generation problem can be distinguished from conventional texture synthesis in three main ways: First, a wide variety of terrain types can be characterized by a combination of global features (such as ridges and valleys) which can be reliably extracted from input terrain maps. In contrast, no such easily identifiable global features exist for general image textures. Second, terrain synthesis must be globally controllable in order to be useful for a wide range of applications, and large terrain data sets (often a gigapixel or more) must be searched in order to meet the users' constraints. This stands in contrast to the canonical texture synthesis problem of "growing" a small input patch of texture into a larger output image with the same local intensity structure. We demonstrate that feature-based terrain matching can address both of these concerns. Third, many recent cut-and-merge techniques for texture synthesis exploit the fact that image textures contain many natural boundaries that provide good seams along which to merge texture elements. In contrast, there are no natural seams in terrain data. Moreover, any mismatch between the height fields in two adjacent patches is immediately visible. To remove height differences by blurring results in highly-visible artifacts. We show how to combine recent graph-cut and Poisson seam removal techniques to address the problem of merging terrain patches.

2.3 Feature extraction

In this section, we describe our method for extracting terrain features from the example height field and from the user's sketch map. We use the term *terrain features* to mean large-scale long curvilinear features such as a river, a valley or a mountain ridge. Such terrain features characterize the overall layout of the terrain. Fig. 3(a) shows an example height field, consisting of a portion of the Grand Canyon, which is depicted as a shaded relief map. This example illustrates both the existence of large scale curvilinear structures (the main river bed) as well as a rich array of secondary structures (the side canyons and other features). Our goal is to match the primary structures to a desired sketch shape while preserving the rich detail that is present in this data.

(a) (b) (c)

Figure 3: Comparison of feature extraction methods. (a) Grand Canyon height field displayed as a shaded relief map. (b) Features extracted by the Canny edge detector. (c) Features extracted by PPA.

It may seem that terrain features could be identified through edge detection. Closer scrutiny, however, reveals several problems in applying edge detection to complex terrain data. First, terrain features are characterized by the local extrema of the height field, while edge detection methods are based on locally-maximal derivatives of the image. As a consequence, the application of edge detection to terrain data results in spurious features due to local height variations, as shown in the upper right

corner of Fig. 3(b).[2] In addition, terrain data has characteristic branching structures which are not always handled correctly by standard edge detection algorithms (e.g. lower right corner of Fig. 3(b)).

We draw upon work in the geomorphology literature to identify large-scale terrain features. In particular, we have adapted the Profile recognition and Polygon breaking Algorithm (PPA), developed by Chang et al. [18], to our task. We briefly summarize the method in this section, and refer the reader to their paper for additional details.

PPA can extract either ridges or valleys. Here we will only describe ridge finding, with the necessary modification to finding valleys being understood. First, each grid point (pixel in the height field) is visited to determine if it is a candidate for being on a ridge. The grid points in eight outward paths are examined (the yellow points in Fig. 4(a)), and the current point is marked as a candidate if the height dips below the central point by more than a threshold[3] along any of these paths. All such candidates are then connected by a segment to all other adjacent candidates (the red circles in Fig. 4(b)). When one segment crosses another, the lower elevation segment is canceled (shown as dotted segments in Fig. 4(b)). The polygons are then broken into dendritic line patterns by repeatedly eliminating the least important segment (the remaining segment with the lowest height, shown as dotted segments in Fig. 4(c)) for each closed polygon. This process terminates when there are no more closed polygons. After the polygon breaking process, shorter branches are eliminated (dotted segments in the Fig. 4(d)).

The output of PPA for a sample terrain is shown in Fig. 3(c). Notice that the valley axes coincide with human perception of the important curvilinear features in the height field. Our system uses PPA to identify features both in the example height field and in the user's sketch map. The gray-scale values in the sketch map

[2]Searching for edges at multiple scales cannot easily solve this problem, as perceptually important terrain structures are not guaranteed to persist over scale.

[3]We typically set the threshold to one half of the elevation range of the input height field.

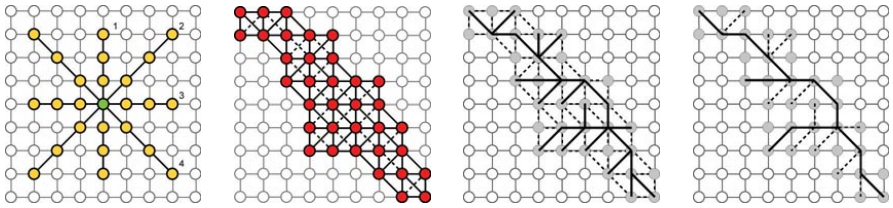


Figure 4: Chang's Profile recognition and Polygon breaking Algorithm (PPA) (a) Profile recognition (b) Target Connection (c) Polygon breaking (d) Branch reduction

are treated as elevation values. The output of PPA is a collection of line segments that are connected and that form long chains along ridges or valleys. These chains of segments are the basis for feature matching between the user's sketch and regions in the given height field. Once the chains of segments have been identified, they are analyzed to form two classes of features: isolated features and curvilinear (path) features. *Isolated features* are branch points and end points. *Curvilinear features* are long chains of segments that connect isolated features. The feature matching process described next make use of these two feature categories.

A key property of the PPA algorithm is that the extracted features form a tree (a forest in general) since all closed polygons are broken in the analysis process. This property allows us to use tree traversal to order the placement of patches during synthesis, as described next.

2.4 Feature-based patch matching and placement

Our synthesis process creates a new height field by extracting patches from the example height field and placing them in an output height field in a manner that is dictated by the user's sketch. Typical patch sizes are 80×80 pixels (this is determined by the spatial scale of the example terrain data and the detail of the result desired by the user). Patch selection and placement is performed in two stages: *feature patch matching and placement* and *non-feature patch placement*.

The first stage, feature matching and placement, locates both isolated features: branch points and end points, and non-isolated features: curvilinear features (*paths*) in the sketch map. See Fig. 5 for examples of each of these feature types. Patches are found that match these features, and these patches are placed in the output height field.

If we treat the isolated features extracted by the PPA algorithm as nodes and curvilinear features (paths) as edges of a graph, then this graph is guaranteed to be acyclic. Our algorithm follows a breath-first-search order to match and align the patch regions. It first picks an isolated feature (usually a branch if it is present) as the root, it then traverses down through the graph one edge at a time until every edge reachable from the root is covered. This process is illustrated in Figs. 6(c)–6(e). Note that this is a departure from traditional texture synthesis methods that follow a rigid patch placement order (e.g. left-to-right and top-down).

Finally, all the as-yet unfilled regions in the output height field are filled using regions from the input that do not contain any strong features. The final result is depicted in Fig. 6(f). In both stages, several candidate patches are considered for placement, and the patch that is selected for use is the one that gives the least cost in terms of its match with the user's sketch and its overlap with already placed patches. (Patch placement invokes a graph cut algorithm to select which pixels should be deposited in the output height field, and this will be described in Sec. 2.5.1). In the following sections, we describe these matching and placement operations in more detail.

2.4.1 Feature Patch Matching and Placement

As described above, isolated features in the user's sketch are either branch points or end points in the segment graph. Fig. 6(b) contains three branch points and five end points. The system analyzes the user's sketch map and identifies all such isolated

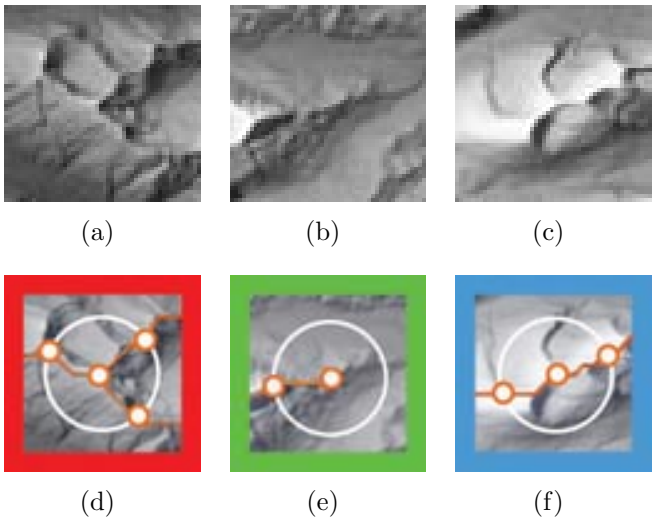


Figure 5: Examples of terrain patches for each feature type: (a),(b),(c) branch point, end point, and path patch respectively. (d),(e),(f) corresponding patches after feature extraction.

features. One by one, each such isolated feature is examined and a list of candidate matches is formed from isolated features in the example height field. We will consider branch points and end points in turn.

Branch Points For branch point matching, the degree (or valence) of the branch points must match and the angles of the outgoing segment chains must be similar. For determining the quality of angle matching, only d possible alignments have to be considered for a degree d branch point. For instance, as shown in Fig 5(d), the degree 3 branch patch would have 3 alignments $(1, 2, 3) \rightarrow (1, 2, 3), (2, 3, 1), (3, 1, 2)$ (The mirror image of this patch is treated as a new patch and its alignment is a separate process.) Because there are typically not many such branch points in a given example height field, testing against all candidate matches is extremely fast.

To perform warping of a candidate patch to fit the user's sketch, first a set of control points $\{P_i\}$ must be identified for the patch. These control points consist of the location of the branch point itself, plus those places where each outgoing path intersects a circle that is inscribed in the patch (see Fig. 5(d)). For instance, a

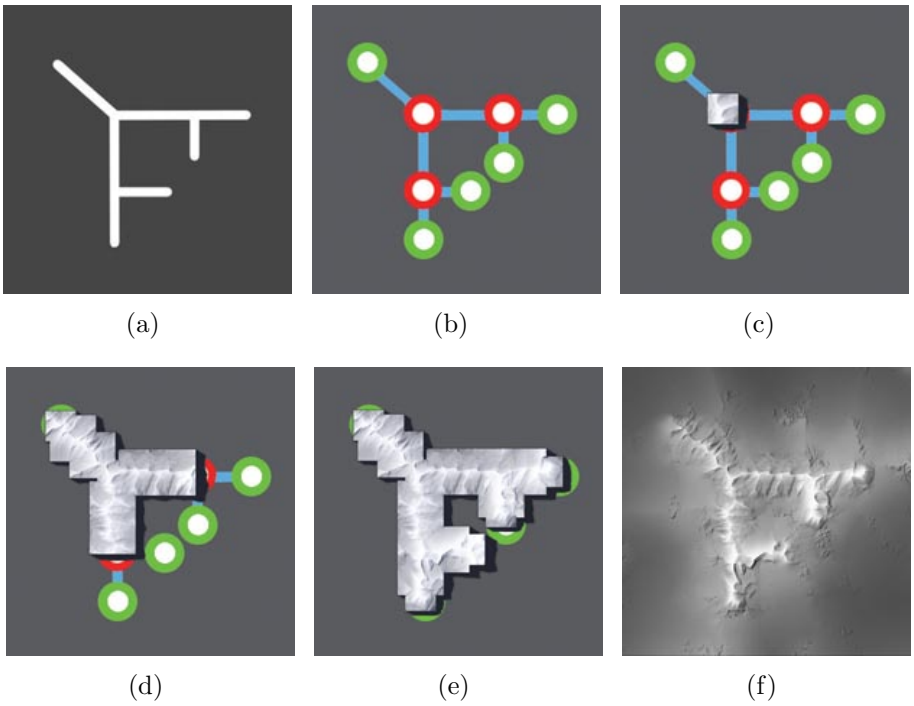


Figure 6: Illustration of patch placement order. (a) Sample sketch map. (b) Tree structure returned by PPA analysis. Branch point features and end point features are connected by curvilinear path features. (c) The root patch is placed first. (d) Breadth-first traversal guides the placement of additional patches. (e) Once tree traversal is complete, begin placing non-feature patches. (f) Final result.

patch with a three-way branch will have a total of four control points. Corresponding control points $\{P_i'\}$ are also defined for the patch from the sketch map. Now we desire a continuous coordinate transformation that maps the control points $\{P_i\}$ to $\{P_i'\}$ and that gives the minimal amount of distortion. We use a thin-plate spline (TPS) interpolant [9] for this purpose because it works well with few constraints and it introduces minimal distortion as measured by the integral bending norm. We use two separate TPS functions to form a coordinate transformation that maps any position in the original height field to its interpolated location in the warped height field. The best k branch point patches with the lowest deformation energy are the candidates for further matching.

End Points Endpoint matching is straightforward because all end points have

similar curvilinear features: a relatively short segment chain going out of the end point, which can be aligned easily. Thus we select all end point patches (Fig. 5(e)) as candidates for further matching without applying a warping in this case.

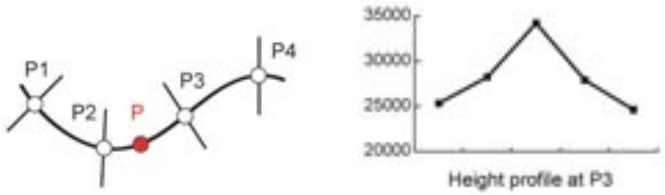
Path Features All of our curvilinear features are chains of line segments (from PPA) that stretch between two isolated features. Our system finds matching patches along such a curvilinear feature while traversing these chains of line segments in the user's sketch. The system travels along these chains in steps that are one-half a patch in size, laying down a patch with each step. As with branching patches, the candidate patches along a curvilinear feature are deformed to better fit the user's sketch. There are always three control points P_0, P_1, P_2 in the candidate patch (Fig. 5(f)), and three corresponding points P'_0, P'_1, P'_2 in the user's sketch that determine the warp. The outer control points P_1 and P_2 are located where the path crosses the inscribed circle of the patch. The central point P_0 is the midpoint in the chain within a patch, which is analogous to the central control point in a branch patch.

All of the candidate patches are ranked according to a combination of matching criteria. The best candidate patch is merged in the output height field. In our current implementation, we use the following matching criteria:

- c_d: Deformation energy from the TPS warping (We refer the readers to [9]) the candidate patch terrain structure matches the sketch map constraints. Although TPS warping can warp the patch into desired configuration in most cases (Except for degenerate cases such as when three control points are collinear), large deformation results in noticeable distortion and is penalized here.

- c_g: Graph cut score (See Appendix A). The graph cut seam cost is an indication of how well the candidate patch matches the already merged patches in the overlap regions (Graph cut automatically handles the case where one patch overlaps with multiple patches and the overlapping can be disjoint). A higher

graph cut score means the seam is more noticeable and harder to remove by Poisson editing.



(a) (b)

Figure 7: (a) Height profiles perpendicular to the path are stored at uniformly sampled points along the path. The height profile at the point P which joins two paths is linearly interpolated from the profiles stored at $P2$ and $P3$. (b) An example of the 5-point height profile that is stored at $P3$.

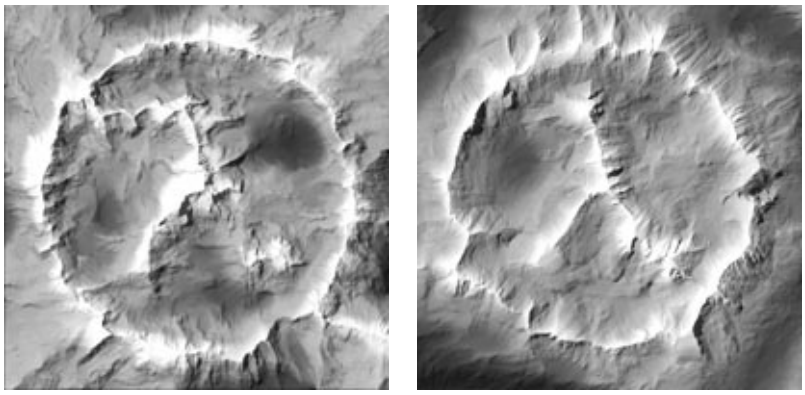
- c_f: Feature dissimilarity cost. The dissimilarity cost measures how closely features along the joining paths are matched[4]. We uniformly sample the curvilinear path and store a coarse height profile perpendicular to the path (our current implementation uses 7-point profile) at each sampled point (Fig. 7). When the path is joining the other path at point P, the height profile at P is linearly interpolated from the profiles stored at its two nearest neighbors, $P2$ and $P3$. Height profiles from joining paths are compared using sum of squared differences.

- c_i's : Other user specified constraint(s) such as height constraint or path smoothness constraint.

The total cost is then a linear combination of all of the matching costs $c = \alpha_d c_d + \alpha_g c_g + \alpha_f c_f + \sum_{i=1}^{n} \alpha_i c_i$. Note that the α's serve not only as weights but also as normalization coefficients for the costs. In our current system, we set $\alpha_d = 1000$, $\alpha_g = 1$, and $\alpha_f = 3$ for most test cases. Changing these coefficients changes the

[4]The graph cut cost gives an overall measure of the overlap quality, but it weights every pixel in the overlap region equally. It is difficult to integrate the feature dissimilarity measure into the graph cut algorithm; therefore, a separate feature dissimilarity cost term is introduced into our system to place emphasis on features.

emphasis on the matching criteria and results in different synthesis results. After curvilinear feature matching and breadth-first-search order placement, the output height field looks like Fig. 6(e).



(a) (b)

Figure 8: (a) Synthesis result using raster-scan patch placement order (b) Synthesis result using tree traversal

Fig. 8 shows a comparison between raster-scan and tree-ordered patch placement using the sketch map of Fig. 10(a) and the example height map of Fig. 10(b). The inner structure of the λ symbol is garbled in the raster scan result (Fig. 8(a)). This is because the row based scan locks onto the boundary of the circle immediately, creating constraints on future matching that prevent good matches in the interior. As Fig. 8(b) shows, both the λ and the circle can be reproduced if the most difficult element (the center branch point) is matched first.

2.4.2 Non-Feature Patch Placement

Once the isolated features and curvilinear features have been placed, the empty areas in the output height field are those places without strong features. A feasible way to fill these areas is to copy patches that match the pixels that have already been placed. To do this, our system "grows out" from the already filled-in areas. Specifically, Square patch positions in coarsely-spaced increments (e.g. every 100 pixels

horizontally and vertically) are placed down in descending order according to the area of the overlapping region with the already synthesized height field.

The system looks for a high-quality match between such a patch and the already synthesized height field based on the sum of squared difference on the overlapping region and selects k candidates. It then finds the best match according to the combination of the SSD score and the graph cut cost in merging. The best patch is placed, and the system continues traversing the output height field looking for overlaps. The process terminates when all pixels in the output map have been filled.

The SSD-based search to find the patch that best matches the already synthesized output height field can be accelerated using Fast Fourier Transforms [90, 56]. Using the FFT-based approach, the matching cost can be computed in $O(n \log(n))$ time, where n is the number of pixels in the source height field. This is in comparison to $O(n^2)$ time for a naive SSD implementation.

2.5 Patch merging

Our system combines two techniques to assure smooth transitions between patches that are placed in the output height field. The first of these is the graph cut technique [58, 10], which finds good seams between already-placed pixels and the pixels from a region that is in the process of being placed. The second procedure solves a discrete Poisson equation [77] to create more gentle transitions between the existing pixel elevations and the pixels from a newly placed patch.

2.5.1 Graphcut Optimal Seam Finder

The graph cut algorithm finds a seam in the overlapping region between patches that determines which pixels will be kept in the final image. This is accomplished by solving a max-flow/min-cut graph problem that minimizes the cost of mismatched elevations across the cut. Edges in the graph represent connections between adjacent pixels, and they are given weights based on elevation differences. For some patches, we

(a) (b) (c)

Figure 9: Illustration of patch placement and seam removal. (a) Matching patches are identified via graphcut. (b) Patch placement results in a seam. (c) Poisson seam remover yields final output.

want to insist that particular groups of pixels from a given patch should be included, such as the central pixels in a branch feature patch. This is accomplished by setting some of the edge weights infinitely high.

As mentioned earlier, the cost of the final seam from the graph cut algorithm is used by the system to select among candidate patches for representing a given feature. Details on the use of the graph cut algorithm for seam finding can be found in [58].

2.5.2 Poisson Seam Remover

After the system has placed a new patch into the output map using graph cut, a discontinuity may still be visible along the seam (Fig. 9(b)). We further improve elevation matching across a seam by adjusting the heights according to artificially set gradient values at the seams. We do this by solving a Poisson equation, similar to the manner in which Pérez et al. perform pixel matching across seams in a color image [77].

Our elevation adjustment stage first translates the elevation values in the overlap region (typically one third of the patch size) into gradient values. Then the gradient values across the seam are artificially set to zero. Finally, a Poisson equation is solved to find the best-fit set of elevations to these adjusted gradient values. The result of this process is a new set of elevations that is much more smooth at the seam (Fig. 9(c)). Our system performs such elevation adjustments locally, invoking the

Table 1: Input DEM locations, resolution and sizes.

Origin	Resolution	Size
Mount Jackson, CO	10m	1620×1620
The Grand Canyon	30m	4080×2760
Mount Vernon, KY	10m	1200×1200
Flathead Range,MT	10m	4097×4097
Puget Sound, WA	10m	4097×4097

method one time per placed patch. Performing elevation adjustment locally avoids the challenge of solving huge matrix equations. For more details, we refer the reader to Appendix B.

2.6 Terrain synthesis results

We used DEMs from U.S. Geological Survey terrain data for the results shown in Figures 1, 10, 11, 12, 13, and 15. The origin, resolution, and size of the data are listed in Table 1.

Our example height fields range from 1200×1200 to 4097×4097 samples at a height resolution of 16 bits. These large terrain maps pose significant search problems for a traditional texture synthesis approach. At our patch size of 80×80 pixels, there are approximately 16 million possible matches in a given 4000×4000 map. However, as a result of our feature-based analysis, we can filter this total set down to approximately 600 match evaluations for a given candidate position to be filled, which dramatically reduces the computational cost. Our method can synthesize a representative terrain in approximately 5 to 6 minutes on an Intel P4 2.0 Ghz processor with 2GB memory. All of the results are rendered using Planetside's Terragen terrain rendering system with procedural textures (determined by height and slope) overlayed on top of the terrain geometry.

Fig. 10 and Fig. 11 illustrate the variation in output of that can be obtained from the same sketch using our terrain synthesis approach. Fig. 10(d) shows the synthesis result (1000×1000) for the Mount Jackson terrain, and the sketch map for this

example is shown in Fig. 10(a). Although this result was created from many patches, both the interior λ and the outer circle are formed by unbroken mountain ridges. Also note the characteristic curtain-like folds in the sides of the mountains. Fig. 10(h) shows the synthesis results (2000×2000) using the Grand Canyon terrain. The sketch map in Fig. 10(e) used for this example is the inverse of Fig. 10(a) so that valleys are selected instead of mountains. Here the λ and the circle are formed from joined pieces of the Colorado River, yet the seam locations cannot be detected. Although guided by the user's sketch, this terrain retains the rich water-carved features of the original data. Fig. 11(d) shows the synthesis result (1000×1000) for the Mount Vernon terrain and Fig. 11(h) shows the synthesis results (2000×2000) using the DEM from Flathead National Forest mountain region. Fig. 12(d) shows the synthesis result (1000×800) for the Mount Vernon terrain with a user sketched Chinese character for "water". These results illustrate the ability of our system to work with a wide variety of terrain types with very different characteristics, while maintaining the salient features of the input sketch map. Fig. 15 provides close-up views of the Grand Canyon and Flathead range synthesis results. Fig. 13(a) shows a rendering of the Grand Canyon height map. We extract the canyon feature from this map using PPA to obtain a sketch map. By synthesizing the sketch in the style of Puget Sound DEM, we obtain a "mountain" version (4000×2000) of the Grand Canyon, illustrated in Fig. 13(b).

Synthesis with Multiple Terrain Styles A larger example, illustrated in Fig. 14, shows the generation of a 3D map of Middle Earth using several different terrain styles. A simple sketch map (Fig. 14(b)) was created by the user from an artist rendering of the map of Middle Earth (Fig. 14(a)). The user assigned different terrain styles to each part of the sketch map (Table 2). In this example, the user first placed down Mount St.Helen at the location of Mount Doom in Mordor (a localized feature). Our system then synthesized the rest of the terrain (3470×2996) automatically and merge it seamlessly with Mount Doom. Note that the user sketch

Table 2: Middle Earth regions

Middle Earth region	Elevation map
Mordor	Puget Sound, WA
Gondor, Rhun, Ered	Rocky Mountain, CO
Moria	Mount Jackson, CO
Dead Marshes	Mount Vernon, KY
Jron	Flathead Range, MT

includes both a variety of mountain ranges and a river valley (the Dead Marshes, near the North-West corner of Mordor). The coast, lakes, and rivers were created using a simple mask, which was extracted from the original map. The synthesized height map (Fig. 14(c)) was rendered using Planetside's Terragen terrain rendering system with both procedural and masked textures to give it an artistic feel. This example illustrates the ability of our system to combine multiple terrain styles into one synthesized terrain and to blend seamlessly among them.

A movie (DivX encoded) showcasing synthesized terrain and additional results can be found at `http://www.howardzzh.com/research/terrain/`.

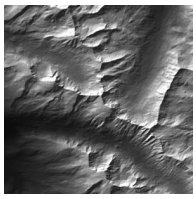
2.7 Conclusion

We have demonstrated that example-based texture synthesis methods can be successfully adapted to the domain of terrain synthesis. The result is a new level of visual realism in user-controllable synthesized terrain. Our approach leverages the fact that useful terrain features can be extracted from height fields using analysis techniques from the geomorphology community.

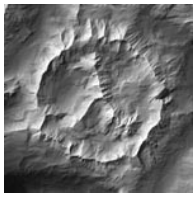
We have introduced a tree-ordered patch placement algorithm which is based on a breadth-first traversal of a feature tree. Our results demonstrate that this placement method is superior to standard raster-scan placement orders.

We have demonstrated the ability to synthesize terrain in widely-differing styles while retaining control over the positioning of major terrain features. Our system

(a) User sketch



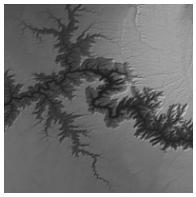
(b) Mount Jackson



(c) Synthesis Result

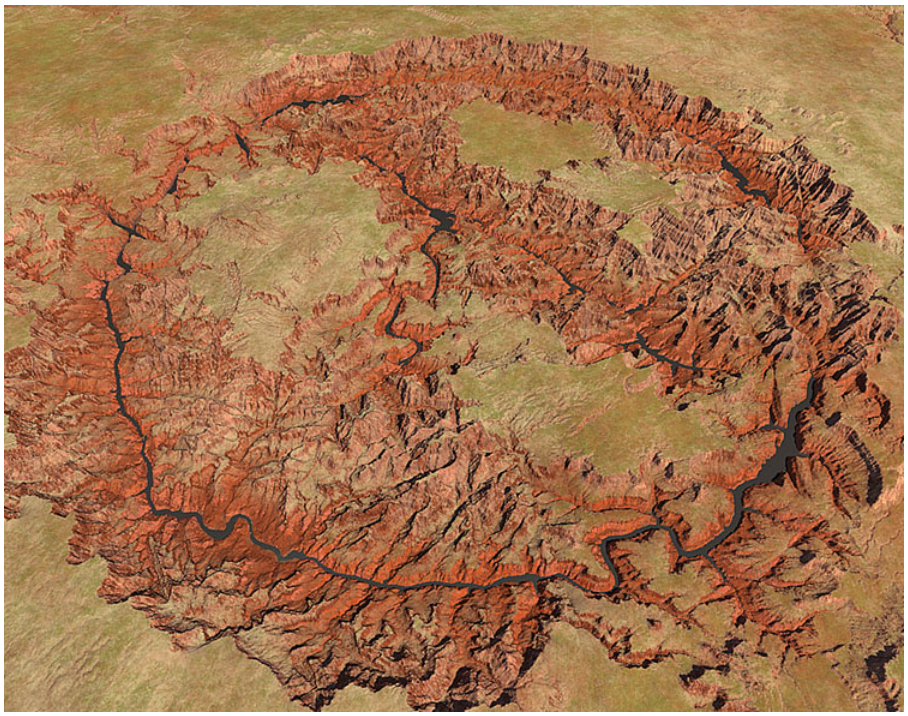
(d) Rendered terrain synthesized from an elevation map of Mount Jackson, CO.

(e) User sketch



(f) Grand Canyon

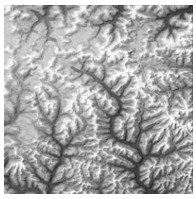
(g) Synthesis Result



(h) Rendered terrain synthesized from an elevation map of the Grand Canyon.

Figure 10: Multiple terrain synthesis results with sketched half-life symbol.

(a) User sketch

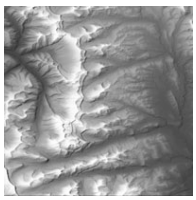


(b) Mount Vernon

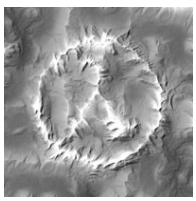
(c) Synthesis Result

(d) Rendered terrain synthesized from an elevation map of Mount Vernon, KY.

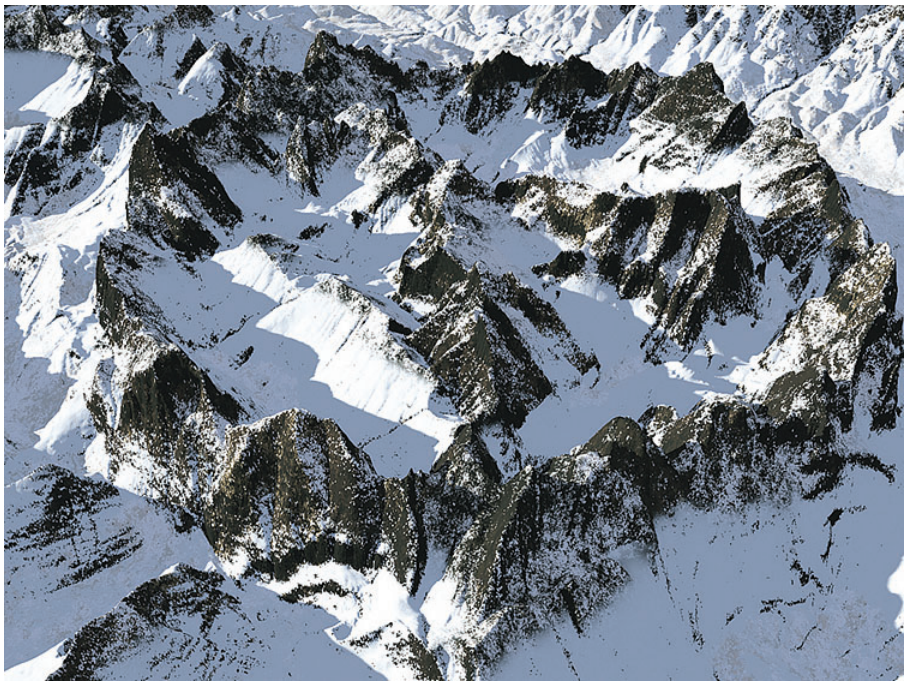
(e) User sketch



(f) Flathead Range



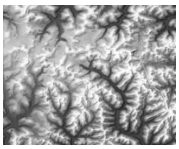
(g) Synthesis Result



(h) Rendered terrain synthesized from an elevation map of Flathead National Forest mountain range, MT.

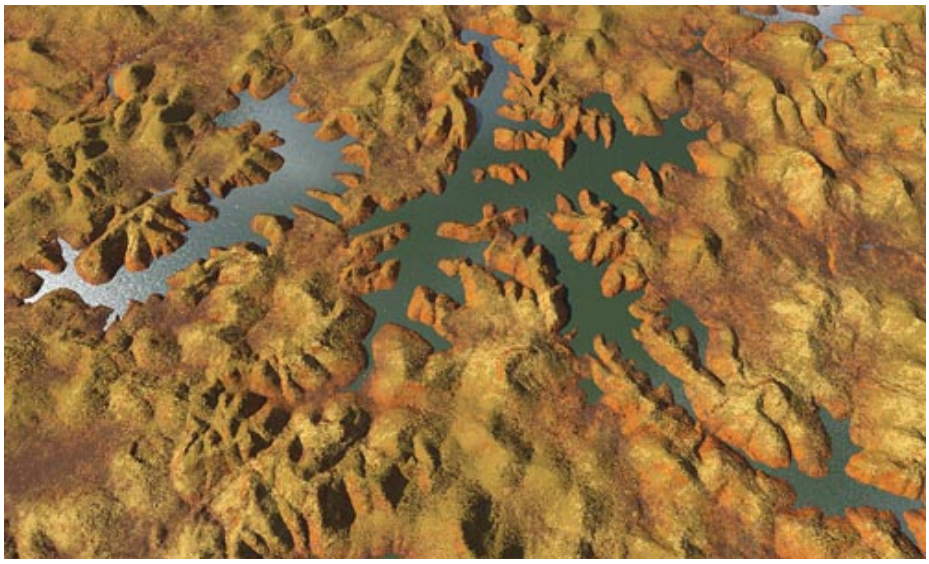
Figure 11: Multiple terrain synthesis results with sketched half-life symbol.

(a) User sketch



(b) Mount Vernon

(c) Synthesis Result



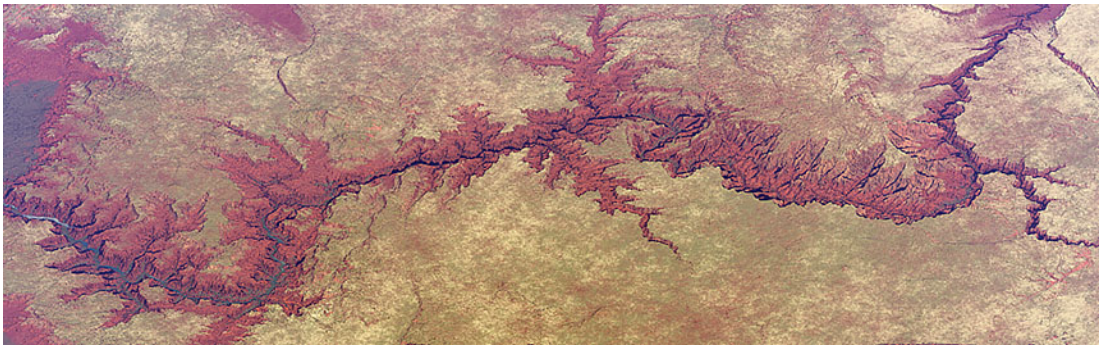
(d) Rendered terrain synthesized from an elevation map of Mount Vernon, KY.

Figure 12: Terrain synthesis result with sketched Chinese character for "water".

is based on an intuitive sketch-based interface for specifying desired terrain characteristics. We believe ours is the first system to make full use of an example-based approach in the domain of terrain synthesis.

Our method suffers from the same limitations as all example-based methods. In particular, the quality of the final synthesis depends upon the richness of the available terrain data. If the terrain features desired by the user cannot be found in the example height field, then it will not be possible to produce the desired result. For instance, our method will perform poorly if the terrain data is from a desert region where few significant curvilinear features are present. The method can also perform poorly when the curvilinear feature pattern is extremely complicated. An additional issue is the need to specify the patch size, which depends in turn upon the resolution and scale of the example terrain.

In the context of this thesis, our approach to exemplar-based terrain synthesis



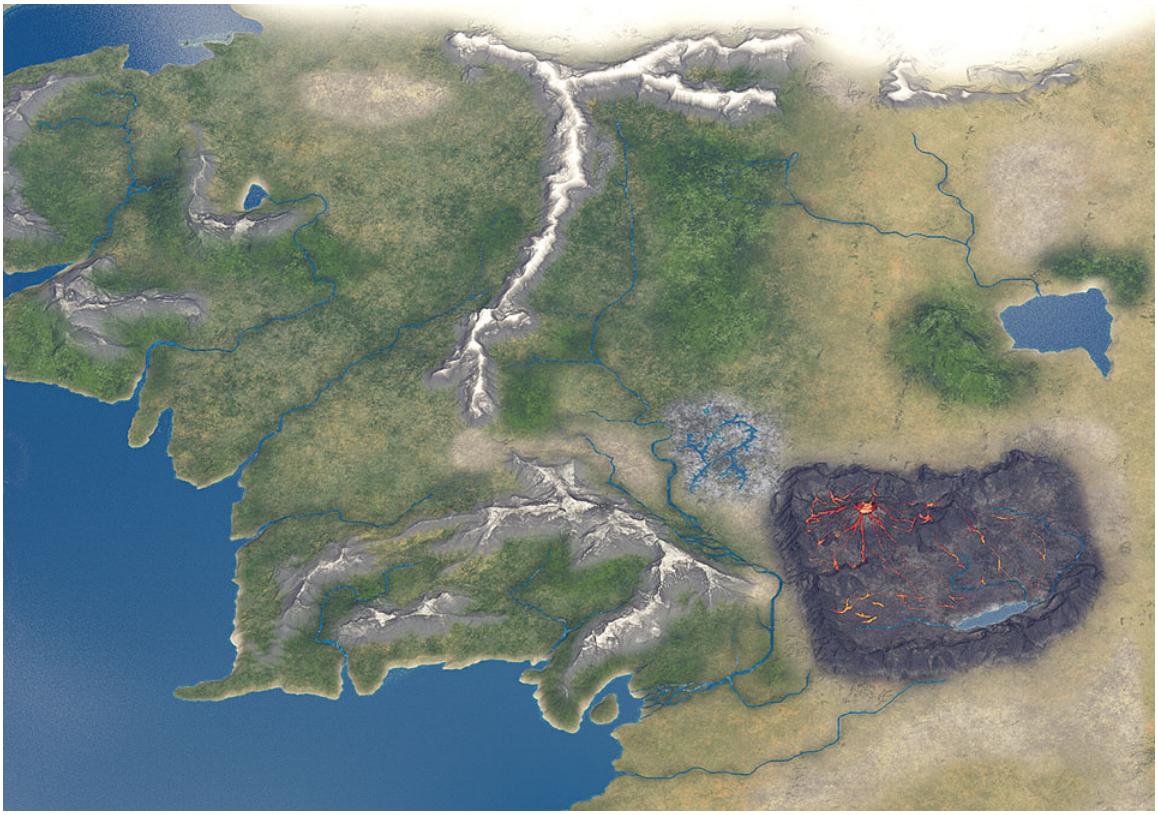
(a)

(b)

Figure 13: The Grand Canyon turned into a mountain range. Features extracted from the Grand Canyon DEM are used as the user sketch to synthesize a mountain range following the structure of the Grand Canyon from an elevation map of the Puget Sound style mountain range.

leverages the restricted domain of terrain DEMs by exploiting their structured properties (e.g. ridges and valleys) in the design of matching and compositing methods. In the following chapters, we will illustrate that the structural properties of dermatological images can be leveraged in a similar manner for the purpose of exemplar-based analysis.

(a) Map of Middle Earth (b) User sketch with masked water system (c) Synthesis result



(d) Rendered terrain, synthesized from multiple elevation maps.

Figure 14: A 3D map of Middle Earth synthesized from multiple elevation maps.

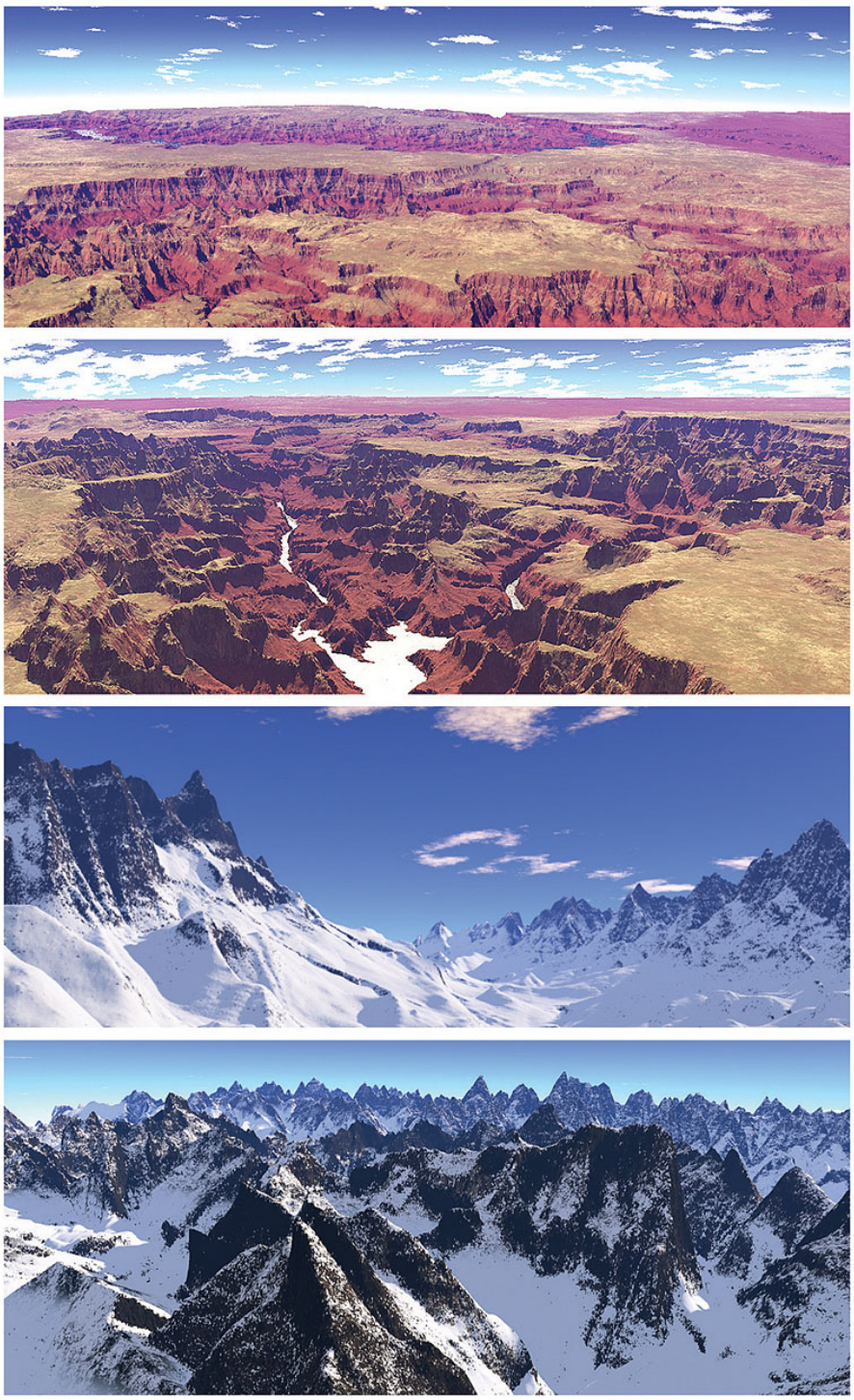


Figure 15: Close-up views of synthesized terrain from the Grand Canyon (top images) and Flathead National Forest mountain range. (bottom images).

CHAPTER III

DERMFIND : A CONTENT-BASED IMAGE RETRIEVAL PLATFORM INCORPORATING EXEMPLAR-BASED IMAGE ANALYSIS

The previous chapter shows that the exemplar-based approach yields substantial benefits for user-guided terrain synthesis. This success is realized through exploitation of structural properties of terrain DEMs. Our goal is to demonstrate that exemplar-based approaches can also yield significant benefits for *image analysis* if we can similarly exploit the structural properties of these images.

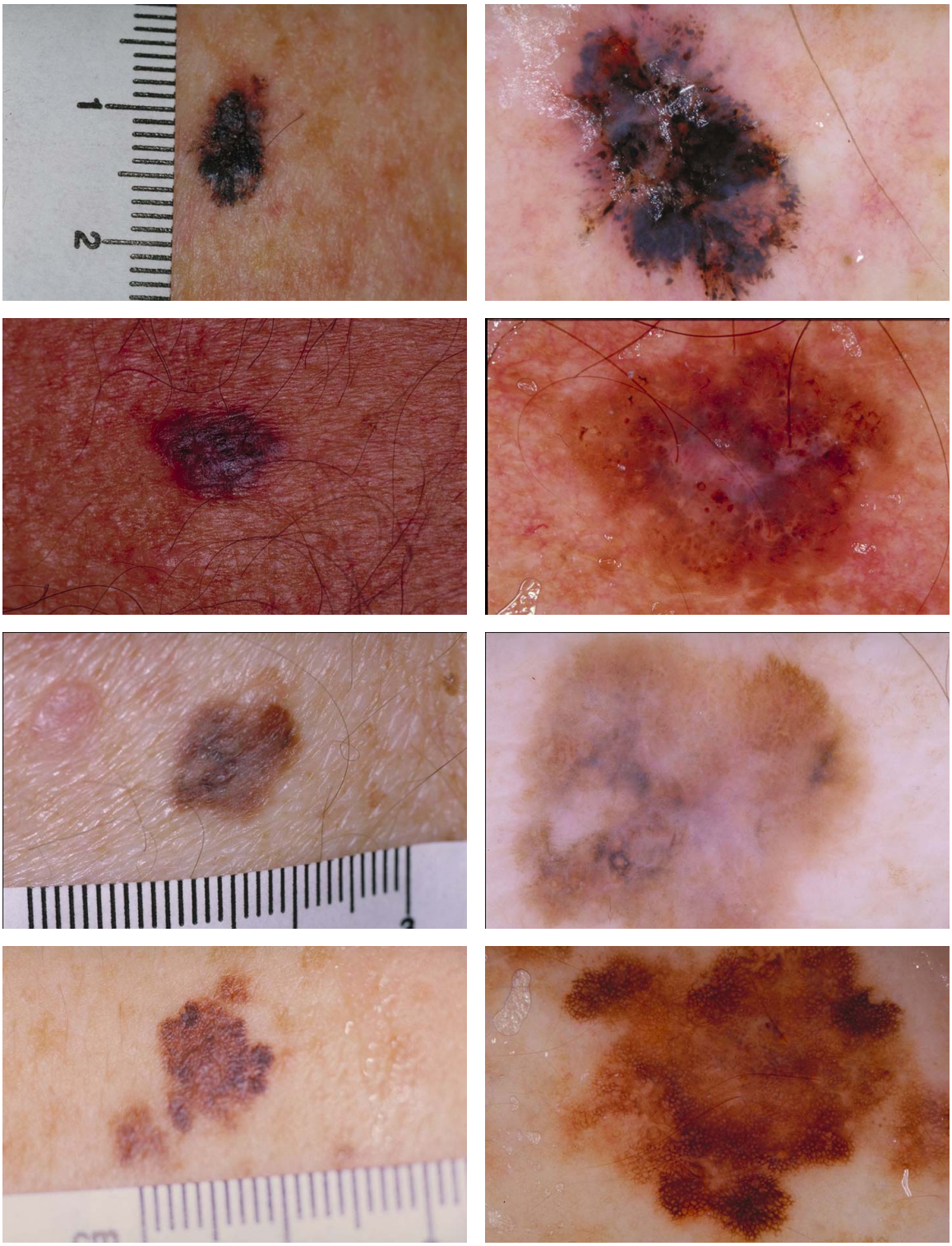
The specific context for our work is the analysis of dermatological images for the problem of content-based image retrieval (CBIR). We demonstrate that CBIR is well suited to an exemplar-based approach. We are motivated by a particular CBIR application: Computer-Assisted Diagnosis of dermoscopy images, an application which is both technically-challenging and medically-important.

3.1 Introduction

Skin cancer is the most common form of malignancy that occurs in the United States (one in two cancer cases) [50]. Worldwide, about one in three cancer cases is skin-related. Among all the skin cancer variations, melanoma, the third most common type of skin cancer, accounts for 75% of skin-cancer caused deaths in the United States. By World Health Organization (*WHO*) estimates, $132,000$ cases of malignant melanoma ($66,000$ deaths) and more than two million cases of other skin cancers occur annually around the world, with 1.1 million in the U.S. alone. These figures are on the rise, and the *WHO* expects the skin-cancer epidemic to accelerate [50].

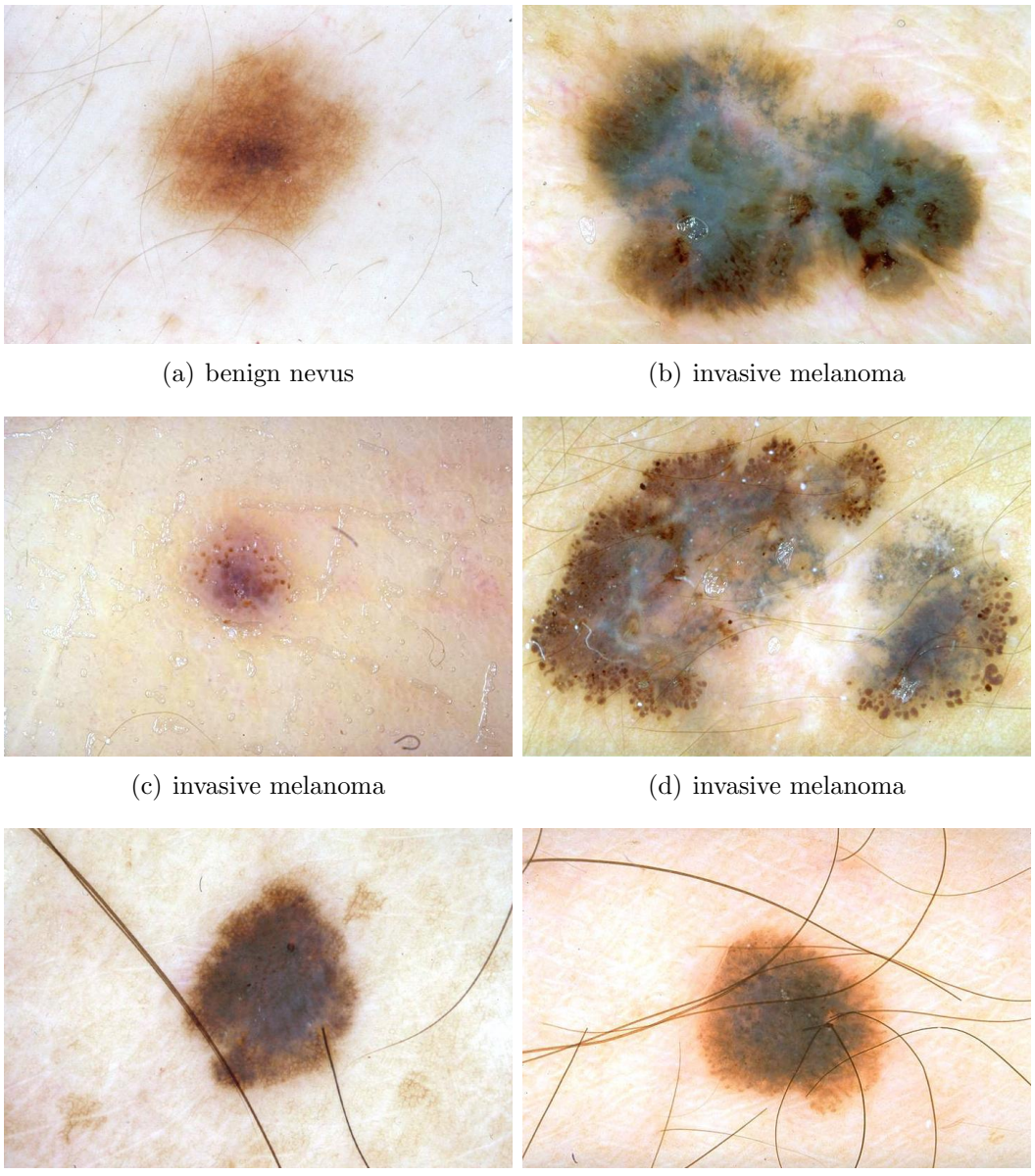
Prevention aside, early detection is the most effective defense against melanoma. Dermoscopy, also known as surface microscopy, is a non-invasive imaging technology that has been shown effective in supplementing the traditional clinical diagnosis of skin cancer. The technique consists of placing a liquid medium on the skin lesion that is subsequently inspected using a hand-held scope with a trans-illuminating light source and standard magnifying optics (a.k.a. dermatoscope, Figure 21(a)). The fluid placed on the lesion eliminates surface reflection and renders the cornified layer translucent, thus allowing a better visualization of pigmented structures within the epidermis which cannot be discerned by the naked eye, as illustrated by the side-by-side comparison in Figure 16. Studies have demonstrated that dermoscopy could provide 10-27% higher sensitivity than unassisted clinical examination when diagnosing melanoma [1]. However, this improvement is only seen among trained clinicians, and in the hands of inexperienced dermatologists, dermoscopy can lower the diagnostic accuracy [1].

The challenge lies in the vastly different appearance of many types of PSLs. Figure 17 consists of example dermoscopy images containing various PSLs. Figure 17(a) shows a typical benign nevus and (b) a malignant melanoma. The pair of lesions shown in Figure 17(c) and (d) have very different appearance yet are both malignant melanoma. In contrast, the lesions shown in Figure 17(e) and (f) look similar, but the left lesion carries favorable prognosis while the right one is a case of highly malignant invasive melanoma. In order to make an accurate diagnosis, an experienced dermatologist often relies on the identification of important dermoscopic features in the lesion. These features range from various pigmentation and global patterns, to local structures such as ridges, blobs, and streaks. A PSL typically exhibits several of these features, and there are over one hundred of dermoscopic features in total (Examples of common dermoscopic features are shown in Figure 18). My work is based on the idea of leveraging the knowledge and experience of trained experts



Clinical view Dermoscopic view

Figure 16: Clinical vs. dermoscopic view of pigmented skin lesions (Image taken from [72], used with permission). Each clinical image was acquired by taking a picture of the lesion in question using a standard camera.



(a) benign nevus

(b) invasive melanoma

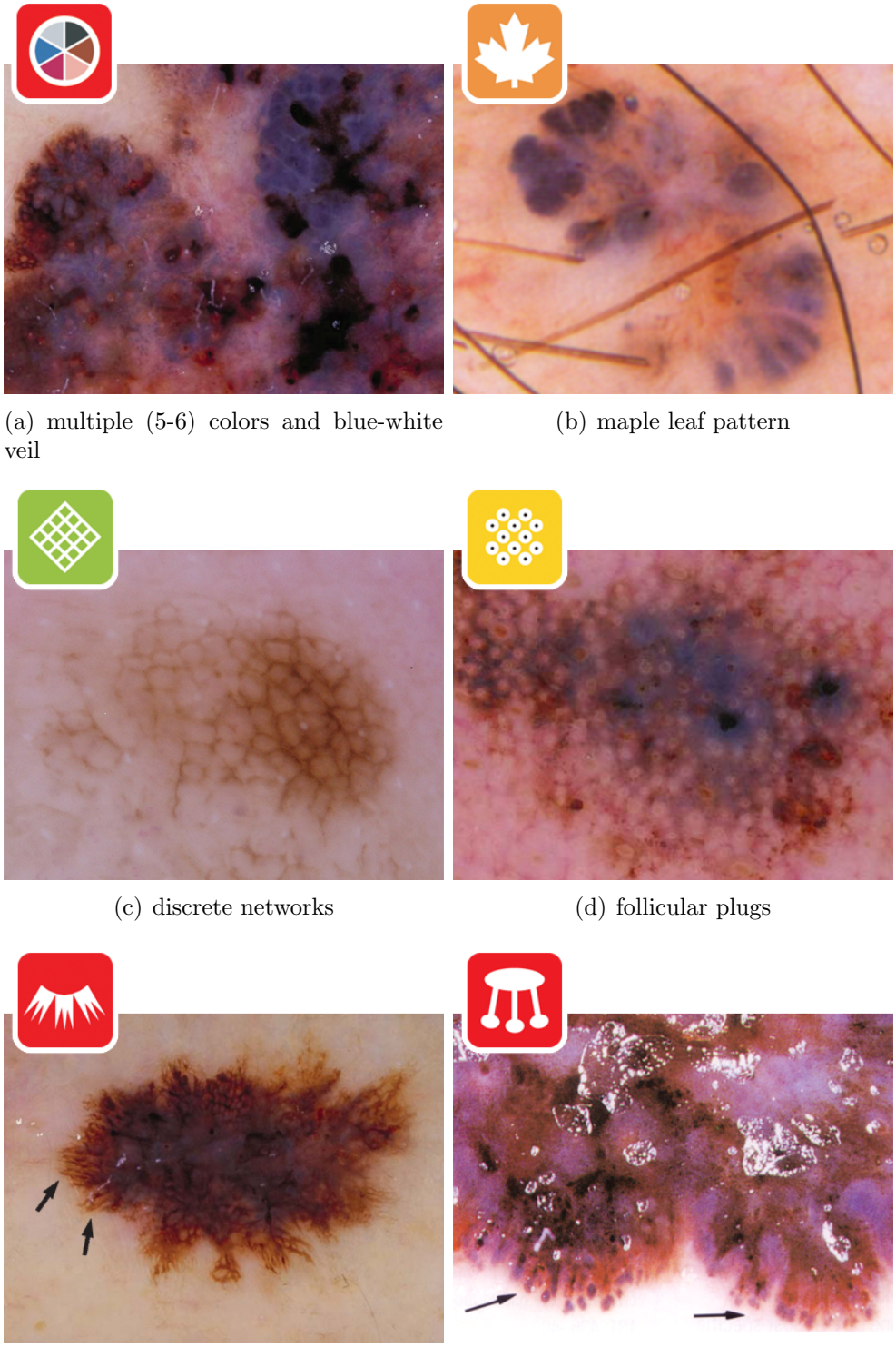
(c) invasive melanoma

(d) invasive melanoma

(e) junctional nevus

(f) invasive melanoma

Figure 17: Example dermoscopy images containing various PSLs with corresponding pathological diagnoses (Image taken from [1], used with permission).



(a) multiple (5-6) colors and blue-white veil

(b) maple leaf pattern

(c) discrete networks

(d) follicular plugs

(e) radial streaming

(f) pseudopods

Figure 18: Examples of common dermoscopic features (Image taken from [72], used with permission)

in dermoscopic image interpretation, so that more people, physicians and patients alike, can obtain access to high-quality diagnostic information. To support this goal, we have developed *DermFind*, [1] the first end-to-end search-assisted decision support system for skin lesion diagnosis. DermFind supports image-based search in a large annotated repository of dermoscopy images. In essence, it is a content-based image retrieval platform that we use to incorporate exemplar-based image analyses, which I will discuss in detail in the following chapters. Given an input image, DermFind can identify similar images based on dermoscopic features and retrieval functions. Meta-data associated with the retrieved images, such as diagnoses and case histories, can be used to support decision-making. DermFind is designed with the realities of clinical practice in mind. It provides a flexible query mechanism and supports interactive refinement.

In developing DermFind, we have made the following contributions:

- DermFind is the first end-to-end CBIR-CAD for search assisted decision support for skin cancer diagnosis,

- It houses one of the largest annotated dermoscopy image repositories which is essential for employing exemplar-based analysis. With a content-based interactive image retrieval system at its core, DermFind serves as a testing platform for our exemplar-based image analysis techniques,

- DermFind utilizes an architecture for efficient search based on hierarchical and distributed data structures which supports interactive query refinement,

- DermFind enables Region-of-Interest (ROI) retrieval and has a relevance feedback mechanism that facilitates user interaction to improve CBIR-CAD performance.

[1] MA significant portion of my research on DermFind was conducted at the Intel Labs Pittsburgh in close collaboration with Dr. Mei Chen, Dr. Richard Gass, and Dr. Laura Ferris M.D.

In the following sections, I survey the landscape of computer aided analysis of dermoscopy. I then describe the system architecture of DermFind and its four main components: 1) image acquisition and data storage, 2) image preprocessing and feature computation, 3) interactive image search, and 4) user interface. Components 1) and 4) are described at the end of this chapter. The details of Components 2) image analysis and 3) interactive search algorithms are described in the following chapters.

3.2 Related work in DermFind

Previous works in computer-aided analysis of dermoscopy usually target one of two goals: automatic diagnosis of an input image (via pattern recognition), or diagnostic support through the retrieval of relevant examples from a library of cases (via content-based image retrieval). Automatic diagnosis has received significant attention [53, 13, 8, 39]. A representative example is the work of Iyatomi et. al. [53], who developed an Internet-based diagnosis system which uses a neural network classifier. MicroDerm [98] and MelaFind [30] are two commercial products for automatic diagnosis. In comparison, our focus is on diagnostic support through retrieval, as this approach seems most likely to meet the needs of clinical practice and education.

While there are a fairly large number of works on image analysis for dermoscopy, there have only been a few attempts to construct a retrieval system for CAD [81, 14, 89, 20]. Rahman et. al. describe one of the more complete CBIR methods for dermoscopy images in [81] which includes preprocessing, segmentation, feature-extraction, and matching. In comparison, DermFind uses a more extensive collection of features, supports ROI queries, and is designed for efficient search and retrieval. The work of Schmid-Saugeon et al. [89] is on lesion boundary extraction and degree of symmetry quantification, while our focus is on the development of a complete CAD system. In [14], Celebi and Aslandogan describe a novel approach to feature matching based on modeling the human perceptual similarity function. While this approach is

interesting, it has not yet been incorporated into a complete CAD system.

DermFind is differentiated from these previous works in three main ways: First, it is the first complete end-to-end diagnostic support system for dermoscopy, integrating a convenient digital image acquisition mechanism, preprocessing including automatic hair removal and cleanup, automatic lesion segmentation and feature extraction, scalable search and retrieval from large databases, and a convenient graphical user-interface for displaying search results and supporting interactive query refinement. Second, DermFind employs a feature analysis pipeline which is based closely on the analysis process used by doctors and which includes state-of-the-art feature representations. Third, DermFind includes a novel region-of-interest(ROI) retrieval mode which makes it possible to refine search results and explore multiple paths of inquiry for a given lesion. In ROI retrieval, the clinician can identify subareas within the lesion and retrieve image cases from the database which match to the sub-area alone. Using DermFind, a clinician can explore a database of more than 1500 annotated images, currently the largest image collection that has been used for retrieval-based diagnosis in dermoscopy.

3.3 System overview

DermFind was designed to be easily-integrated into the existing clinical work-flow of a dermatology practice. The system is comprised of four main components (Fig 19: image acquisition and data storage, image processing and feature computation, interactive image search, and user interface. Note that these components can be geographically distributed for telemedicine applications.

Figure 19 illustrates how DermFind is integrated into the existing clinical workflow of a dermoscopy practice. Starting like a conventional dermoscopy session, the clinician uses the dermatoscope to exam the patient's lesion. However, our specially designed dermatoscope also captures the image and wireless transfer it to the

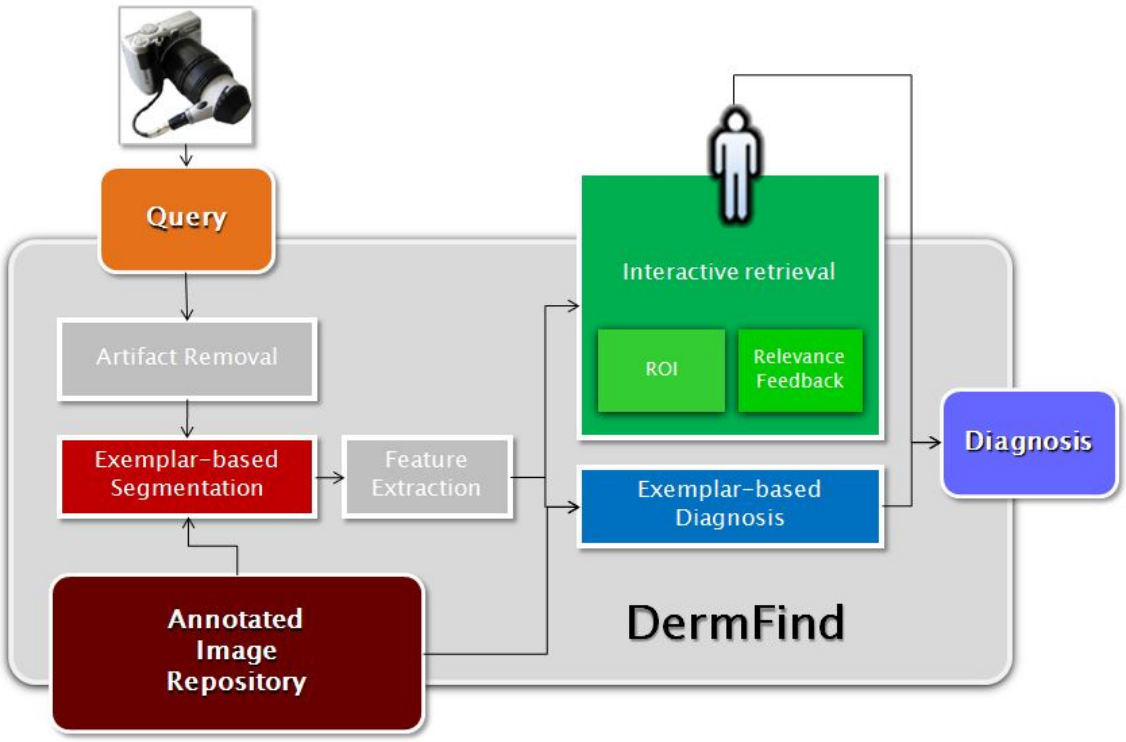


Figure 19: DermFind Overview

DermFind data storage unit. This new image, marked as "Query" in Fig. 19, then goes through a series image processing and analysis: artifact removal (Chapter 4), segmentation (Chapter 5), and feature extraction (Chapter 6). These steps take less than 10 seconds in total to complete. The incoming dermoscopy image goes through this process as illustrated in Figure 20. Afterwards, the clinician will be able to select arbitrary region-of-interest from the incoming lesion image and perform search for similar cases over the entire DermFind image repository (Chapter 7). Due to its efficiently designed search algorithms, DermFind is capable of supporting relevance feedback to the clinicians for interactive refinement. At the end of this pipeline, clinicians can use the meta-data associated with the retrieved images, such as diagnoses and case histories, to support their decision-making.

In the following sections, we will first look at the image acquisition and user interface components.

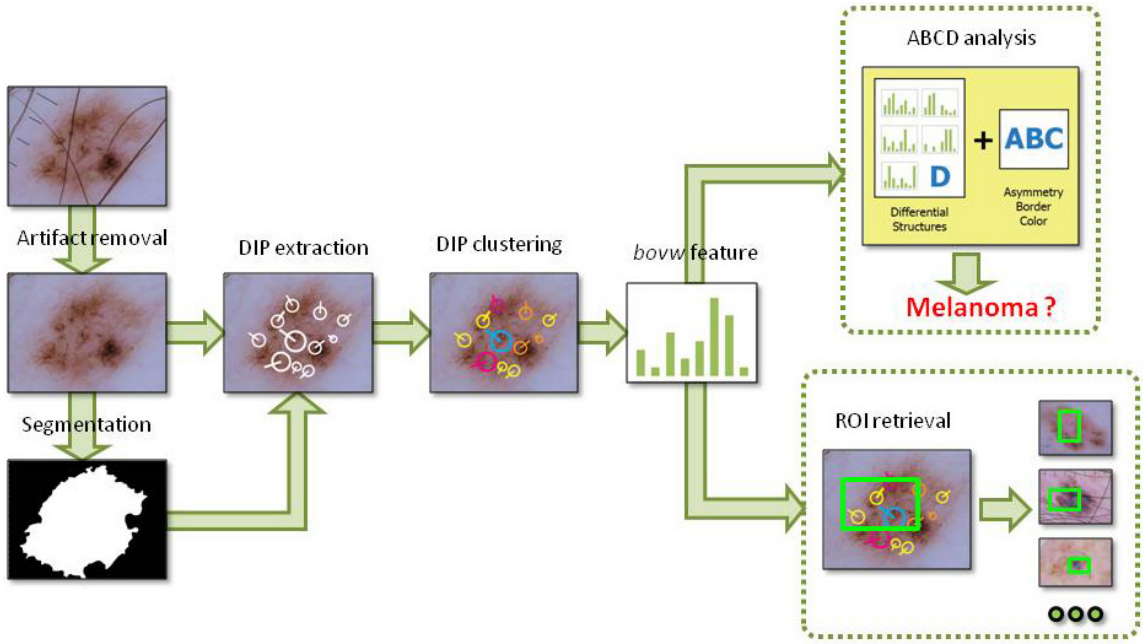


Figure 20: Life of an image in DermFind

3.3.1 Image Acquisition and Data Storage

In conventional practices, clinicians mount cameras on the back of dermatoscopes, taking pictures as they exam the patients. These dermoscopy images are uploaded only after the session ends, stored for archiving or future research purpose. In contrast, we facilitate integration into the clinical work-flow with a custom-designed digital dermatoscope, DermCam, depicted in Fig. 21 (c), comprising a Heine Delta 20 dermatoscope (Fig. 21 (a))and a Canon PowerShot A570 IS digital camera (Fig. 21 (a)). Dermatoscopes typically come with adapters that can be attached to a camera (in our case, through a Canon LA-DC52G, 52mm Conversion Lens Adapter) to capture the image data. However, we notice that in the 5-10 minute per patient reality of the U.S. clinics, even the bulk of the two small devices and the act of affixing them with a snap can be bothersome. This motivated us to design our own device. We remove the heavy cylindrical battery pack of the Heine dermatoscope and power the unit from the camera's battery, making the unit compact and maneuverable. We use the Eye-Fi SD card [32] to store and wirelessly transfer the images to the DermFind

Table 3: Sources of images stored in DermFind case repository.

Dataset source	cases	images	Notes
UPMC	308	309	UPMC Dermatology Department
Dermoscopy	217	217	Book:'Dermoscopy' (CD)
IDS	928	1022	Web: International Dermoscopic Society
EDRA	1011	1047	Book:'Interactive Atlas of Dermoscopy'(CD)
NYU	65	65	Web: NYU Medical Center Interactive Tutorial
Misc	4	4	Web: Misc images
Total	2533	2664	

system [2]

The transmitted images are detected by the Eye-Fi software running on the DermFind system and saved into a user configured directory. DermFind automatically detects that new images are uploaded DermFind automatically detects the arrival of a new dermoscopy image and initiates pre-processing, which removes image artifacts (using the method described in Chapter 4) and detects the lesion boundary (detailed in Chapter 5). Lesion characteristics which are relevant to skin cancer diagnosis such as color, shape, texture, and dermoscopic features are automatically extracted (detailed in Chapter 6). This initial processing takes less than 10 seconds, enabling clinicians to perform image search immediately after the image acquisition procedure.

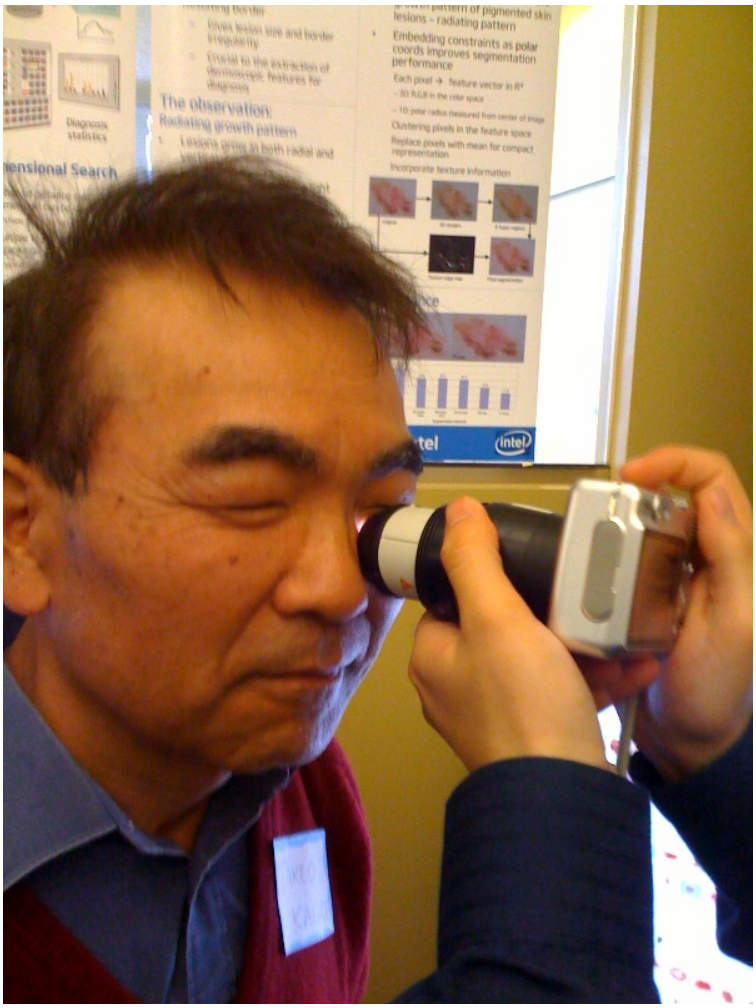
Once processed, the images are stored in an image repository that has a relational SQL database at its core, which is illustrated in Fig. 3.3.1(a). Currently DermFind stores a total of 2533 dermatological cases, among which 1570 have confirmed pathological diagnoses. The distribution of the diagnoses of these cases are shown in Table 3.3.1(b). These images were collected from various sources including clinical images from the Department of Dermatology at University of Pittsburgh Medical Center, images from companion CD-ROMS of various books, and images from websites and online forum, as listed in Table 3.3.1.

[2]The camera unit was designed by Dr. Aaron Cois and Dr. Richard Gass at Intel Labs Pittsburgh.

(a) Heine Delta 20 dermatoscope

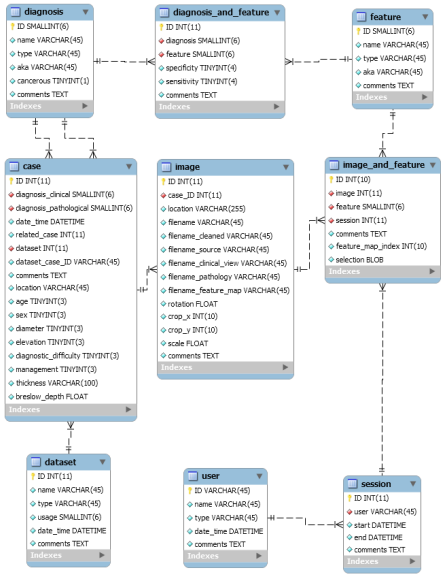
(b) Canon PowerShot A570 IS digital camera

(c) DermCam: Wireless digital dermatoscope



(d)

Figure 21: a) Heine Delta 20 dermatoscope b) Canon PowerShot A570 IS digital camera c) DermCam: Wireless digital dermatoscope d) DermCam in action: This picture was taken during a live demo session of the DermFind system at 2008 Intel Labs Pittsburgh Open House. The subject in the picture is Professor Takeo Kanade from the Robotics Institute at Carnegie Mellon University.



Number of	Number
cases (images)	2533 (2664)
with diagnosis	1570 (1617)
malignant	495
benign	1075
melanoma	405
basal cell carcinoma	68
squamous cell carcinoma	6
nevi	265
dysplastic nevi	599
Seborrheic keratoses	94
vascular lesion	62
misc	39
macule	32

Figure 22: a) The design of the DermFind database. b) The diagnosis distribution of all the cases in our database.

Because images have different properties associated with them (e.g. with ground-truth pathological diagnosis) and came in as batches at various algorithm development stages, we detail the various datasets used in this thesis in Table 3.3.1

3.3.2 Graphical interface for user interaction

DermFind is an interactive application and much effort has been devoted to the graphical user interface. Feedback from clinicians was used to fine-tune the visual display

Table 4: Dataset used in the evaluation of various components of the DermFind system.

ID	Dataset Usage	images
DA1	Artifact removal (Section 4.4)	460
DS1	Segmentation (Section 5.4)	67
DS2		110
DS3		1787
DF1	DIP feature extraction (Section 6.4.1)	150
DE1	ABCD analysis evaluation (Section 8.1.1)	1269
DE2	Retrieval performance evaluation (Section 8.1.3)	1617
DU1	User study (Section 8.2.3)	24

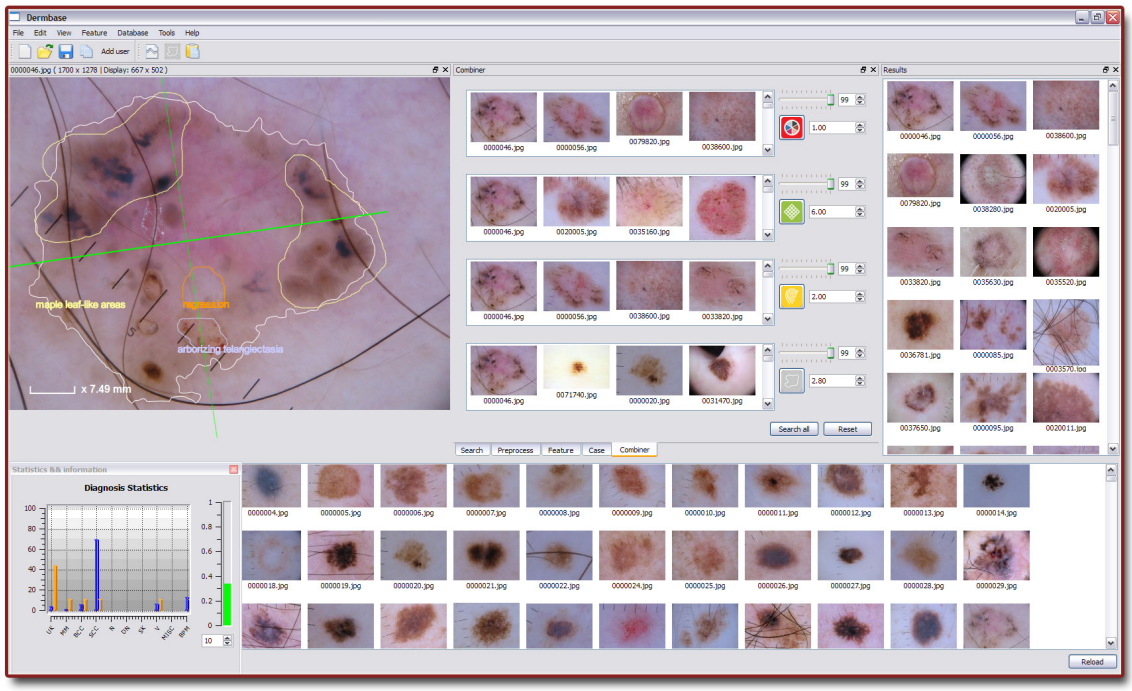


Figure 23: DermFind user interface

and tool design. Fig. 23(right) shows a screen-shot from the interface, which supports browsing into the image repository, provides a toolset for close examination of images (zoom, virtual ruler, color histogram, and editing tools), and enables the user to define search criteria based on one or more image characteristics with a relevance threshold. Much of the design and development of DermFind user interface is centered on the practicality of it being integrated into physicians' daily clinical practices, and I will discuss the details in the interactive search and user study sections.

In the ensuing chapters, I will demonstrate how the DermFind system can be used as a context for developing various exemplar-based image analysis and interactive search techniques, but first, I will address an artifact removal technique which serves as a preprocessing step to clear up the newly acquired images.

CHAPTER IV

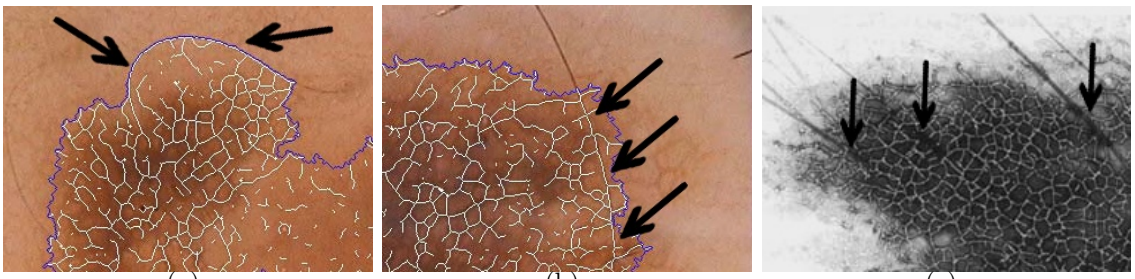
FEATURE-PRESERVING ARTIFACT REMOVAL VIA EXEMPLAR-BASED INPAINTING

Computer procedures designed for extracting and classifying the intricate features demonstrated by pigmented skin lesions in dermoscopy images can be distracted by the presence of artifacts like hair, ruler markings, and air bubbles. Therefore, reliable artifact removal is an important pre-processing step for improving the performance of computer-aided diagnosis of skin cancer. In this chapter, we present a new scheme that automatically detects and removes hairs and ruler markings from dermoscopy images. This method also addresses the issue of preserving morphological features during artifact removal. The key components of this method include explicit curvilinear structure detection and modeling, as well as feature guided exemplar-based inpainting, which preserves underlying lesion features by replacing the regions polluted by artifacts with most relevant clean regions from the image itself through feature matching. Experiments performed on real dermoscopy images demonstrate that this method produces superior results compared to existing techniques.[1]

4.1 Introduction

When acquiring dermoscopy images for computer-aided diagnosis, the practitioners mount cameras on the back of dermatoscopes, taking pictures as they exam the patients. A typical examining session only lasts several minutes. As a result of the time constraint, some common artifacts such as hair and air bubbles often appear in the

[1]The work described in this chapter has been published in [108].



(a) (b) (c)

Figure 24: Artifacts interfering with computer procedures designed to extract dermoscopic features. (a) Hair affecting lesion segmentation (image courtesy of Grana *et al.*). (b) Hair masquerading as pigmented network (image courtesy of Grana *et al.*). (c) Hair interfering with pigmented network extraction (image courtesy of Fleming *et al.*).

acquired images. The presence of such artifacts can interfere with many analytic procedures required for accurate diagnosis, as illustrated in Fig. 24. In the first example (Fig. 24(a)), the segmentation algorithm presented in Grana *et al.* [42] mistakes a piece of hair (indicated by the arrows) as the lesion boundary. In Fig. 24(b), the hair at the top of the image confuses the same network analysis procedure [42] and masquerades as part of the pigmented network. Similarly in Fig. 24(c), hairs break the network pattern extracted by a similar procedure developed by Fleming *et al.* [34].

Therefore, removal of such artifacts is an important prerequisite for accurate computed-aided diagnosis of pigmented skin lesions. Previous efforts in artifact removal from dermoscopy images include the works of Schmid *et al.* [89] and Fleming *et al.* [35]. Schmid and his co-authors proposed a method that detects and removes hair using global morphological operations and thresholding. They observe that in dermoscopy images hair segments are often thin structures darker than their surroundings. Accordingly, their algorithm starts with transforming the dermoscopy image into the *CIE L*u*v** colorspace and extracting thin structures from its luminosity channel using a morphological closing operation (Fig. 27(c)). It then generates

a hair mask by globally thresholding on these thin structures based on their luminosity. At the end, it replaces each masked pixel with an average of its neighboring non-masked pixels via another morphological operation. This algorithm is efficient and the results are good enough for future processing in many cases. It has been adopted by the *DullRazor* hair-removal software [62][2]. However, this global approach can sometimes lead to unsatisfactory results. For instance, because the morphological closing operation essentially replaces each hair pixel with its neighborhood average, it can often leave behind undesirable blurring and color bleeding in the result. Moreover, in some dermoscopy images (Fig. 27(b), the assumption of hair being darker than the underlying skin and lesion does not always hold, and important dermoscopic features can be mistaken as artifacts and removed. To handle such cases, more explicit hair modeling is required. Fleming *et al.* [35] developed an automated hair detection and tracing algorithm which treats hair as long, relatively straight curvilinear structure with constant width and curvature. This algorithm however, does not model intersecting hair explicitly, so its performance degrades on dermoscopy images containing many hair segments.

To address these issues, we present a novel system that automatically detects artifacts with curvilinear property, i.e. hair and ruler markings, and removes them from dermoscopy images.

- The system performs curve fitting and models curve intersection explicitly to improve accuracy on curvilinear structure detection. This minimizes the chance of removing lesion features accidentally.

- The system removes artifacts using exemplar-based inpainting, which preserves underlying lesion features.

Both improved detection accuracy and feature-preserving artifact removal lead to

[2]We use results produced by DullRazor for comparison purpose.

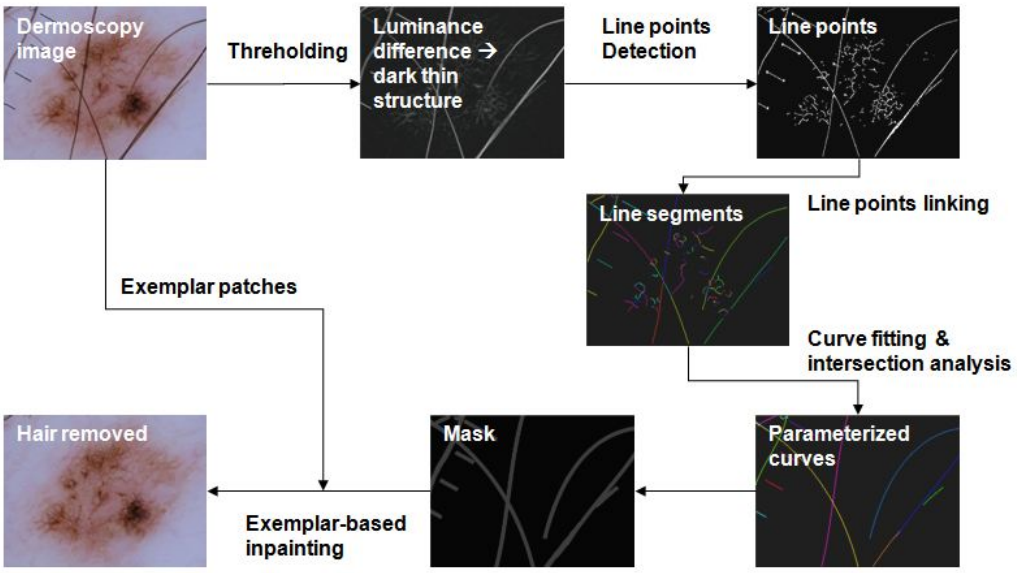


Figure 25: The flowchart describing the work flow of our artifact detection and removal scheme.

superior results compared to existing techniques.

Fig. 25 shows the work flow of our approach. The system takes a dermoscopy image as input and generates a luminance difference image [89] to enhance the thin-dark structures present in the image. It then applies Steger's line detection algorithm [91] at multiple scales to extract line segments with certain widths. An intersection analysis algorithm is also applied at this time to reconnect the line segments broken at intersection. Once short segments are grouped together, the algorithm attempts to robustly fit a curve to each hair segment using RANSAC [33]. Among the good fits, the algorithm accepts long curves, estimates their width, and adds them to the artifact mask. When the mask is complete, it invokes an exemplar-based inpainting procedure [23] to replace the masked sites with its best match from the same image. The replacement patches are selected according to a similarity measure that takes both appearance and feature into account. The procedure stops when all of the artifact pixels are replaced.

4.2 Line detection and curve fitting

4.2.1 Line points detection and linking

Given a dermoscopy image, we first enhance the thin-dark structures following Schmid *et al.*'s approach [89]. The resulting gray level image (shown in Fig. 27(c) with black/white inverted) is the input for the line points detection step. We adopt the Steger's algorithm [91] to detect line points and link them into line segments. The detection algorithm regards the gray level image as a surface in which pixel intensity corresponds to surface height. The line points are points where the first directional derivative in the direction of the line vanishes and the second directional derivative have a large absolute value. A point (x, y) is a line point if it satisfies $(tn_x, tn_y) \in [-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$, where (n_x, n_y) is the normalized eigenvector that corresponds to the maximum absolute eigenvalue of the local Hessian matrix $H(x, y)$[3], and t is evaluated as follows:

$$t = -\frac{r_x n_x + r_y n_y}{r_{xx} n_x^2 + 2 r_{xy} n_x n_y + r_{yy} n_y^2}, \tag{1}$$

where $r_x, r_y, r_{xx}, r_{xy}, r_{yy}$ are partial derivatives of the image estimated by convolving the image with discrete two dimensional Gaussian partial derivative kernels. The standard deviation σ of these kernels is directly tied to the expected line width. Therefore, we apply the line detection algorithm at multiple scales (with different σs) to extract line segments within a certain width range. Fig. 27(d) shows the line points detected from Fig. 27(c). The saliency of a line point (x, y), i.e. the absolute value of the second directional derivative along (n_x, n_y), is inversely proportional to its intensity. After individual line points are identified, we adopt a modified version of Peter Kovesi's [57] linking algorithm to link these points into line segments i.e. sets of ordered points.

[3] (n_x, n_y) points to the direction perpendicular to the line direction at point (x, y)

4.2.2 Curve fitting

As can be seen in Fig. 27(d), the line detection algorithm successfully extracted all the discernible hair segments. However, many thin, dark lesion features were also mistakenly extracted. Since some of these false detection have higher saliency than many true hair segments, thresholding on saliency would not reject the false positives. Moreover, due to image noise and hair color variation, long hair segments are often broken even after the line linking step. This renders thresholding on curve length ineffective too. We observe that individual hairs in most dermoscopy images appear as long, relatively straight curvilinear structures with constant width and slowly varying curvature [35]. In order to reduce false detections, we attempt to fit a curve to each line segment using the least square method [51]. This initial curve fitting step serves mainly as a pre-screener to reject clearly non-hair like structures, i.e. short segments or curves with very high curvature. Let a line segment be given by an ordered set of n points $P_i = (x_i, y_i)$, $i = 1, 2, \ldots, n$. Because the curves we are going to fit tend to be straight, we use the power basis expression for a Bézier curve $B(t) = (B_x(t), B_y(t))$

$$\begin{cases} B_x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \\ B_y(t) = a_y t^3 + b_y t^2 + c_y t + d_y, \quad 0 \le t \le 1. \end{cases} \tag{2}$$

For each given line point $P_i = (x_i, y_i)$, its corresponding parameter value t_i in the Bézier curve power basis expression $B(t_i) = (B_x(t_i), B_y(t_i))$ is calculated using a cord-length parametrization

$$t_i = \begin{cases} 0 & i = 1, \\ \dfrac{\text{length of polygonal line } P_1 P_2 \ldots P_i}{\text{length of polygonal line } P_1 P_2 \ldots P_n} & 1 \le i \le n. \end{cases} \tag{3}$$

The squared sum S of the distances between P_is and their corresponding points $B(t_i)$s on the curve is computed as

$$S = \sum_{i=1}^{n} (\text{distance between } B(t_i) \text{ and } P_i)^2 \quad (4)$$

$$= \sum_{i=1}^{n} (a_x t^3 + b_x t^2 + c_x t + d_x - x_i)^2 + \sum_{i=1}^{n} (a_y t^3 + b_y t^2 + c_y t + d_y - y_i)^2. \quad (5)$$

By setting the partial derivative with respect to each coefficient to zero, we obtain the following system of linear equations for x_i and likewise for y_i. We solve these systems using Gaussian elimination and find coefficients a_x, b_x, c_x, d_x, a_y, b_y, c_y, and d_y that minimize S

$$\begin{cases} a_x \sum_{i=1}^{n} t_i^6 + b_x \sum_{i=1}^{n} t_i^5 + c_x \sum_{i=1}^{n} t_i^4 + d_x \sum_{i=1}^{n} t_i^3 &= \sum_{i=1}^{n} x_i t_i^3 \\ a_x \sum_{i=1}^{n} t_i^5 + b_x \sum_{i=1}^{n} t_i^4 + c_x \sum_{i=1}^{n} t_i^3 + d_x \sum_{i=1}^{n} t_i^2 &= \sum_{i=1}^{n} x_i t_i^2 \\ a_x \sum_{i=1}^{n} t_i^4 + b_x \sum_{i=1}^{n} t_i^3 + c_x \sum_{i=1}^{n} t_i^2 + d_x \sum_{i=1}^{n} t_i^1 &= \sum_{i=1}^{n} x_i t_i \\ a_x \sum_{i=1}^{n} t_i^3 + b_x \sum_{i=1}^{n} t_i^2 + c_x \sum_{i=1}^{n} t_i^1 + d_x n &= \sum_{i=1}^{n} x_i. \end{cases} \quad (6)$$

$$\begin{cases} a_y \sum_{i=1}^{n} t_i^6 + b_y \sum_{i=1}^{n} t_i^5 + c_y \sum_{i=1}^{n} t_i^4 + d_y \sum_{i=1}^{n} t_i^3 &= \sum_{i=1}^{n} y_i t_i^3 \\ a_y \sum_{i=1}^{n} t_i^5 + b_y \sum_{i=1}^{n} t_i^4 + c_y \sum_{i=1}^{n} t_i^3 + d_y \sum_{i=1}^{n} t_i^2 &= \sum_{i=1}^{n} y_i t_i^2 \\ a_y \sum_{i=1}^{n} t_i^4 + b_y \sum_{i=1}^{n} t_i^3 + c_y \sum_{i=1}^{n} t_i^2 + d_y \sum_{i=1}^{n} t_i^1 &= \sum_{i=1}^{n} y_i t_i \\ a_y \sum_{i=1}^{n} t_i^3 + b_y \sum_{i=1}^{n} t_i^2 + c_y \sum_{i=1}^{n} t_i^1 + d_y n &= \sum_{i=1}^{n} y_i. \end{cases} \quad (7)$$

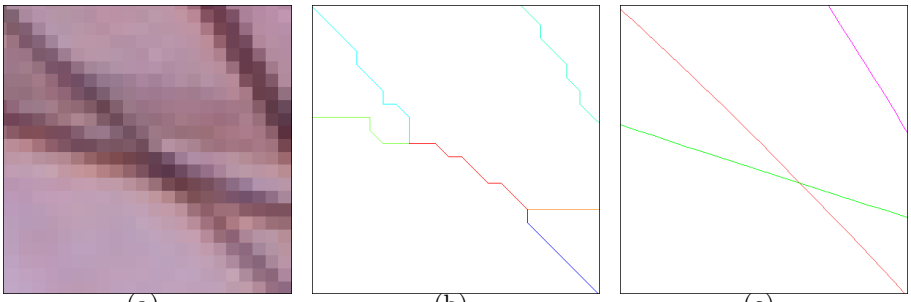
For robustness, we employ the RANSAC [33] algorithm during the curve fitting step.

4.2.3 Linking line segments

In dermoscopy images, hair often appear as long curvilinear structure with relatively constant curvature. Therefore, it is more desirable to model it as a single lone segment rather than a train of short segments. Due to image noise, however, the line point linking step often fails to deliver the former. A further line segment linking step is necessary for improved robustness and detection accuracy. Fitting a curve to each

line segments not only rejects non-hair like structures, it also facilitates this next step. Fig. 26(a) shows a portion of a dermoscopy images with hair artifacts. The line segments extracted after the line point linking step is shown in Fig. 26(b). Notice that the green-orange pair of line segments come from the same piece of hair, but it would be difficult for a computer algorithm to detect by just looking at the ends of these line segments. The robust curve fitting procedure, on the other hand, rejects line points that are outliers. The resulting parametric curve not only assumes sub-pixel precision but also has more accurate tangents at possible connecting sites. As a result, our algorithm can then perform windowed searches to identify curve pairs having ends *meet* (within a close vicinity of each other) and reconnect the pairs with matching tangents at the ends. Fig. 26(a) shows a scenario where the linking procedure described above is not suitable. Two hairs intersect each other at a sharp angle, resulting several broken lines and an odd segment in the middle shared by both hairs (Fig. 26(b)). There are a number intersection configurations like this that can lead to inaccurately linked line segments. Our algorithm detects such intersections by identifying special junction segments such as the red segment in Fig. 26(b). It then looks up to a number of common configurations for the best way to pair up the involved line segments, i.e. green-red-orange and blue-red-cyan in this case. At the end, these pairs of segments are linked and refit. Fig. 26(c) shows the parametric curves resulted from this intersection analysis and line segment linking step.

Having a parametric form for each hair also simplifies its width estimation. At each point along the hair, the width can be calculated by measuring the distance between the two foothills of the pixel intensity profile in the normal direction of the curve. Since hair width mostly stays constant in dermoscopy images, we need only estimate the width at one point on the hair, i.e. the most salient line point. In practice, we take an average along each hair to improve robustness. After obtaining the parametric form and width of each hair, we generate an artifact mask to mark



(a) (b) (c)

Figure 26: Linking line segments at hair-hair intersections. (a) A close-up view at a hair-hair intersection. (b) Line segments extracted after the line point linking step. (c) Parametric curves resulted from the intersection analysis and line segment linking step.

the pixels to be removed from the original dermoscopy image.

4.3 Exemplar-based inpainting for artifact removal

Previous works [89] replaces each artifact pixel with the average of its non-artifact neighbors. This approach often leaves behind undesirable blurring and color bleeding(Fig. 28(c)), which may interfere with analytic procedures required for computer-aided diagnosis. To address this issue, we adopt the exemplar-based inpainting and patch-ordering mechanism developed by Criminisi *et al* [23]. At each iteration of the inpainting process, the target patch to be replaced is selected according to a priority term composed of a confidence and a data term defined at every pixel. The confidence term measures how certain a pixel is an non-artifact pixel; this is computed from the confidence of surrounding non-artifact pixels, or artifact pixels that have been already replaced. The data term captures the *strength of flow* of an edge at each pixel on the artifact region boundary. The confidence term enforces that patches containing less artifacts are filled in first while the data term propagates prominent image structures, i.e. edges, in the filling direction. The target patch selected according to this patch-ordering mechanism is then replaced with a source patch found in the original image. The source patch is chosen to minimize the sum squared error (SSE) from

corresponding non-artifact pixels in both patches. This exemplar-based approach with a fill-in ordering dictated by image structure ensures that missing morphological features inside artifact regions are replaced with already existing features from the same image in a systematic way. Our procedure stops when all artifact pixels are replaced.

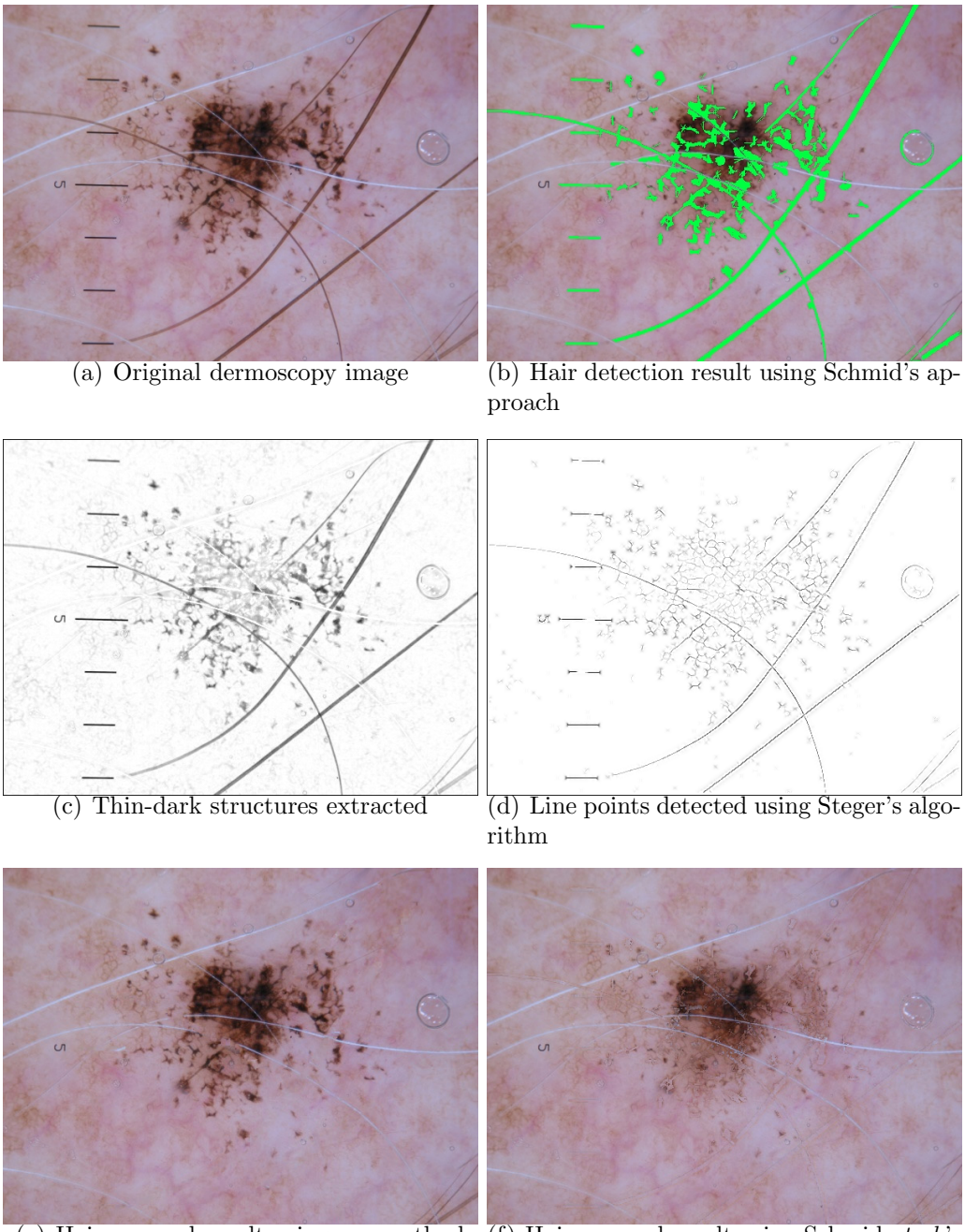
4.4 Experimental Results

We tested our algorithms on dataset DA1 (Table 3.3.1), which includes 460 dermoscopy images. These images are of various qualities. About one eighth of the images have visible artifacts. Our algorithm automatically detects these visible artifacts and removes them. Fig. 27 shows a side-by-side comparison between the artifacts detection and removal results generated by Schmid et al.'s and our method, respectively. Notice that in Schmid et al.'s hair detection result (Fig. 27(b)), although most of the pixels associated with hairs or ruler markings are picked up, a few non-artifact pixels are falsely selected as well due to their dark appearance. We point out that these false detections cannot be avoided by lowering the threshold, because the colors of the skin pigments are almost identical to the hairs. With explicit hair shape modeling, our algorithm successfully detects the hair pixels and rejects dark skin pigments. Fig. 27(f) and 27(e) show artifacts removal results generated by Schmid et al.'s algorithm and ours, respectively. While most of the noticeable hair pixels are removed from both images, the result generated by Schmid et al.'s algorithm has a noticeable blurred appearance at many removed pixel sites. This is due to the interpolation nature of their algorithm. The image generated by our method is free of such artifacts. Upon closer examination of both results (Fig. 28(c) and 28(b)) and the same portion of the original image (Fig. 28(a)), The difference is visible, our algorithm produces results that better preserve the appearance of the original feature. Fig. 29 shows some additional results, ranging from a few isolated hair segments to

hairs with many intersections.

4.5 Conclusion

Artifact removal is an important pre-processing step for computer aided diagnosis of pigmented skin lesions from dermoscopy images. I have presented a novel artifact detection and removal scheme. Automatic hair and ruler marking detection is achieved by curvilinear structure analysis. Explicit curve fitting is performed to increase the robustness of our detection algorithm. After the artifact pixels are selected, they are replaced using feature guided, exemplar-based inpainting, through which, morphological features that are important to diagnosis are better preserved. For dermoscopic images polluted by hair and ruler markings, this method produces visually superior restoration in comparison to existing techniques, and prevents artifacts from affecting the ensuing lesion segmentation and dermoscopic feature detection steps.



(a) Original dermoscopy image

(b) Hair detection result using Schmid's approach

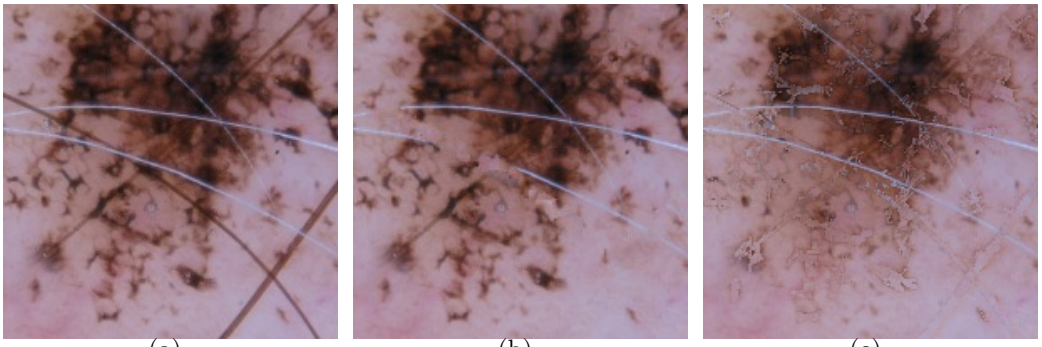
(c) Thin-dark structures extracted

(d) Line points detected using Steger's algorithm

(e) Hair removal result using our method

(f) Hair removal result using Schmid *et al.*'s method

Figure 27: A side-by-side comparison between the results obtained using ours algorithm and Schmid et al.'s algorithm.



(a) (b) (c)

Figure 28: Close-up views of images from the result comparison in Fig. 27. (a) Original dermoscopy images. (b) Result from using our feature-preserving artifact removal method. (c) Result from using Schmid *et al.*'s method.

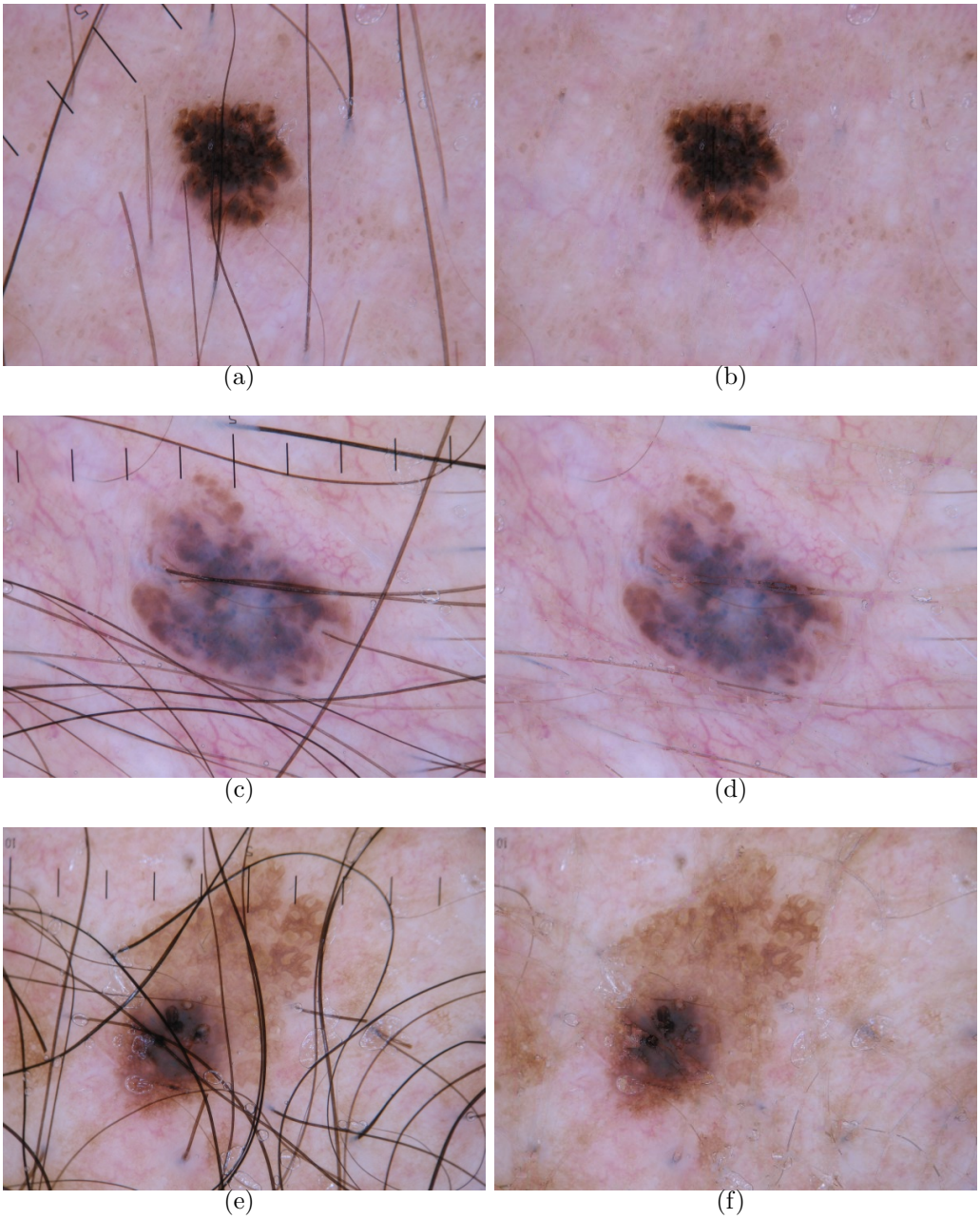


Figure 29: Additional results. (a), (c), and (e) Original dermoscopy images. (b), (d), and (f) Results after performing feature-preserving artifact removal.

CHAPTER V

EXEMPLAR-BASED SEGMENTATION OF PIGMENTED SKIN LESIONS FROM DERMOSCOPY IMAGES

In this chapter, we describe an exemplar-based image analysis technique we have developed in the context of DermFind, a novel exemplar-based algorithm for segmenting pigmented skin lesions (PSLs) in dermoscopy images. The method leverages the context provided by a global color model to retrieve annotated examples which are most similar to a given query image. Pixel labels are generated through a probabilistic voting rule and smoothed using a dermoscopy-specific spatial prior. We compare our method to three competing techniques using a large database with hand-segmented ground truth. Our exemplar-based approach yields significantly better segmentations and is computationally efficient. Preliminary experiments with fundus images of the retina suggest that our exemplar-based approach has promise for other segmentation tasks in medical image analysis.[1]

5.1 Introduction

Computational analysis of dermoscopy images often starts with the segmentation of a lesion from its surrounding skin. Column 2 of Figure. 36 show several examples of lesion segmentation. The resulting border structure provides a basis for the calculation of important clinical features, such as lesion size and symmetry axes. In addition, it is crucial for the extraction of some of the most discriminating dermoscopic features such as radial streaming and pseudopods. In recent years, a number of methods have been developed for the automatic segmentation of PSLs in dermoscopy images.

[1]The work described in this chapter has been published in [110].

See [17] and Sec. 5.5 for a review.

A majority of these methods assume the lesion and skin pixels define two separable distributions in some chosen color feature space (i.e., *RGB*, *HSI*, *CIELUV*, and *CIELAB*), in which case a good segmentation can be found by maximizing the separation between distributions. Unfortunately, this assumption does not hold in practice. Pigmented skin lesions can vary widely in appearance, resulting in lesion color distributions that are not compact and exhibit substantial overlap with skin. Four common sources of variability are depicted in Fig. 36: (1) low contrast between the lesion and surrounding skin (Fig. 36 first row), (2) compound characteristics in lesion appearance, i.e. differences in appearance within the lesion are greater than the differences between lesion and skin (Fig. 36 second row), (3) fragmentation due to regression or depigmentation (Fig. 36 third row), and (4) complex background skin appearance (Fig. 36 fourth row). When methods based on the maximum separation assumption are applied to such images, they are likely to produce erroneous results.

The key idea in our approach is that the contextual information provided by the overall appearance of the lesion and its surrounding skin can be exploited to adaptively adjust the classification criteria on a per-pixel basis. We are inspired by the ability of experienced dermatologists to recognize the various conditions under which a particular pixel could be lesion, based on having seen a large number of examples of PSLs. We describe a novel exemplar-based classification algorithm which indexes into a database of previously-segmented dermatoscope images to identify relevant examples, and then generates segmentation labels through a probabilistic voting method. Our approach produces significantly better results than methods which employ a single global segmentation criteria, and it is extremely fast and simple to implement.

In addition, we show that a technique we developed for enforcing spatial constraints on dermoscopy image segmentations [107] can be utilized to smooth the

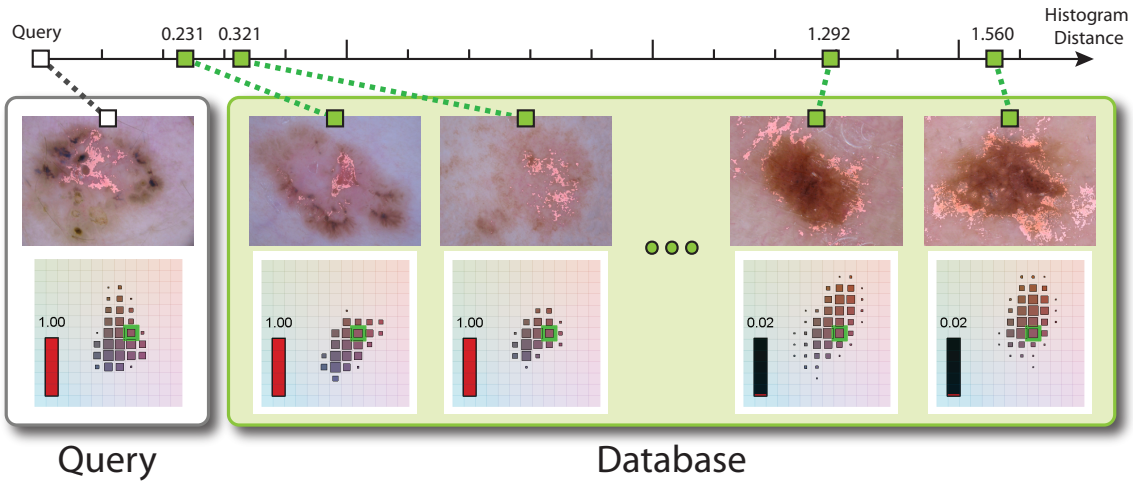


Figure 30: Illustration of exemplar-based segmentation algorithm with adaptive context.

segmentation map, resulting in a more uniform output and an additional decrease in segmentation error. The performance of our approach is compared to existing popular segmentation methods using a hand-labeled dataset of more than 2300 images, which is among the largest in the published literature. Our experiments quantify the benefits of our method in comparison to existing techniques.

5.2 Exemplar-based pixel classifier

Fig. 30 illustrates our exemplar-based classification approach. We start with a gallery of pre-segmented exemplars. Each exemplar has an associated color histogram, which defines the "context" for image matching. First, we identify the neighboring exemplars for a given query image, via chi-squared histogram matching. The exemplars are illustrated in the green box in Fig. 30, ordered by distance. Below each exemplar is a 2D rendering of its color histogram, where the size of each gray square is proportional to the number of counts. For illustration purposes, we show a 2D slice (in α and β channel) of the actual 3D histogram. The size of each gray square is proportional to the number of counts.

Our goal is to classify each histogram bin as lesion or skin, thereby labeling all

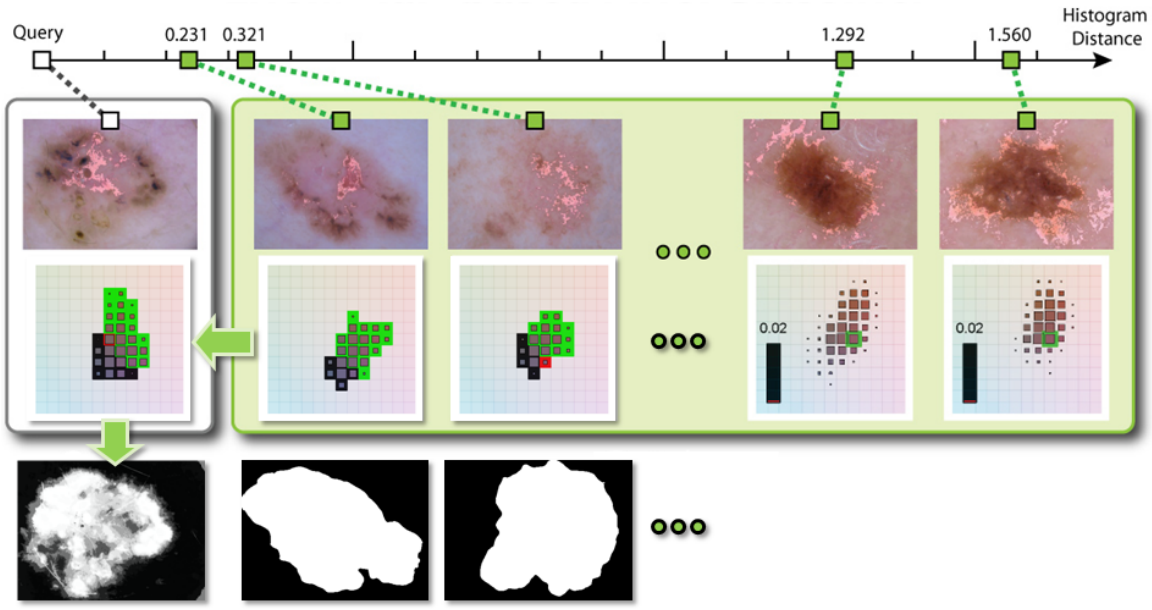


Figure 31: Illustration of exemplar-based segmentation algorithm with adaptive context.

pixels in the query image. A representative bin, whose correct label in the query image is "lesion," is shown with a green border in the figure. The pixels belonging to this bin are highlighted in each of the images. Note that the highlighted bin arises as both lesion and skin across the exemplar set in Fig. 30, underscoring the limitation of the separable assumption. The highlighted bin has a probability of being lesion in each of the exemplars, as indicated by the red/black bars in the figure. These exemplar probabilities are combined using weights derived from the histogram distances to obtain the predicted lesion probability in the query image histogram (in this example, correctly classified as lesion). The leftmost exemplars, being the closest to the query image, determine the vote, while the rightmost exemplars are ignored. Note that his approach is completely general and could be applied to any image-based segmentation problem.

We now describe the approach in more detail. Let $\vec{X} = \{\vec{x}_i\}_{i \in S}$ be the observed data from an input image where S is a set of image sites to be labeled, and $\vec{x}_i \in \mathbb{R}^c$ is the feature vector at site i (in this work, x_i is pixel color and $c = 3$). Let $H(\vec{X})$

denote the histogram of feature values for the image pixels. The corresponding labels are given by $\vec{Y} = \{y_i\}_{i \in S}$, $y_i \in \{0, 1\}$, representing skin or lesion. We define the label probability distribution for pixel x_i as $P(y_i|\vec{h}_m, H(\vec{X}))$, where $\vec{h}_m$ is the color feature vector at the center of the histogram bin m which contains x_i. Note that in this model, spatially-adjacent labels are conditionally independent given the histogram. We discuss the enforcement of spatial constraints in Sec. 5.3.

In addition, we are given a library of exemplar images $\mathcal{X} = \{\vec{X}_j\}, j \in \{1, \cdots, J\}$ with a corresponding set of ground-truth labellings $\mathcal{Y} = \{\vec{Y}_j\}$. We use the annotated exemplars to construct the posterior probability distribution for each quantized feature vector as follows:

$$P_i(y_m|\vec{h}_m, H(\vec{X})) \equiv \frac{1}{K} \sum_{k=1}^{K} w_k P(y_m|\vec{h}_m, H(\vec{X}_k)), \tag{8}$$

where K is some fixed number of closest neighbors. Here, proximity is defined by the histogram distance between two images, $d_j = d(H(\vec{X}), H(\vec{X}_j))$. The normalized weight $w_k = d_k^{-2} / \sum_{i=1}^{K} d_k^{-2}$ determines the influence each exemplar has on the final decision. Given the probability model in Eq. 8, the final label assignment is given by $\vec{Y}^* = \arg\max_{\vec{Y}} P(\vec{Y}|H(\vec{X})) = \arg\max_{\vec{Y}} \prod_i P_i(y_m|\vec{h}_m, H(\vec{X}))$. Returning to Fig. 30, we can identify the histogram distances as d_k. For the highlighted bin m, the red/black bars correspond to $P(y_m|\vec{h}_m, H(\vec{X}_k))$. The pixels corresponding to bin m are rendered in each image with the color $\vec{h}_m$.

Fig. 32 illustrates the segmentation results and associated exemplar neighbors for four representative examples of lesions that are difficult to segment reliably. The four query images are illustrated in column 3. Columns 4-7 show the four nearest neighbors returned from the database. The images in column 2 illustrate the per-pixel lesion probabilities (high probabilities are white) produced by our method. Column 1 shows spatially-smoothed probability estimates produced using the technique described in Sec. 5.3.

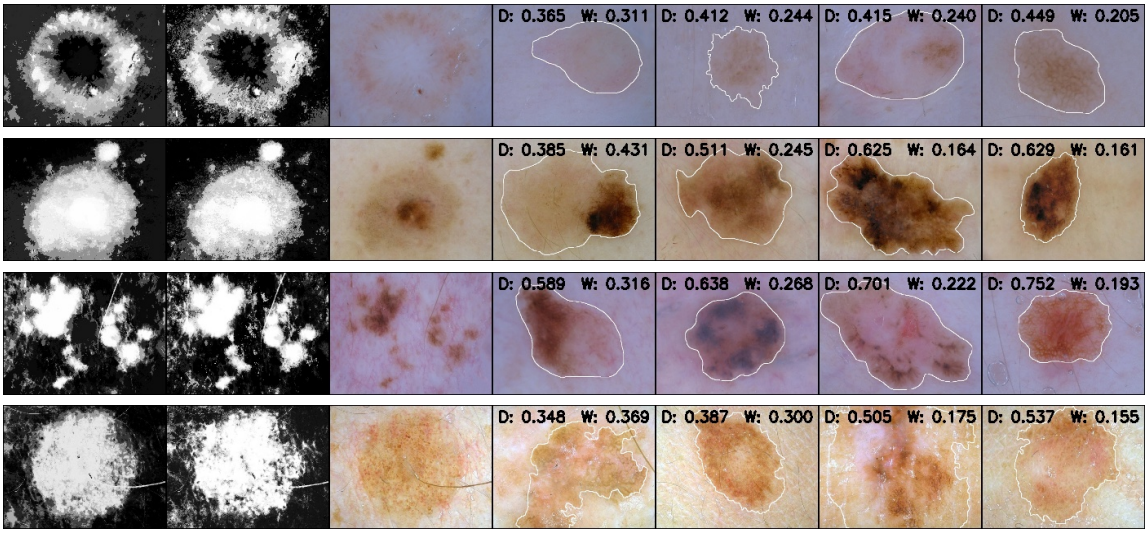
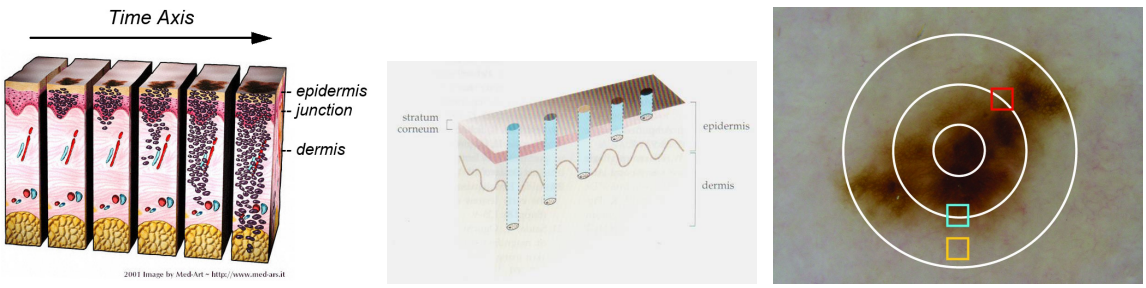


Figure 32: Algorithm performance on four difficult example lesions (one per row). Col 2 is the posterior probability map, in which brigher pixels have a higher probability belonging to the lesion. Col 1 is the result after enforcing dermoscopy specific spatial constraints on col 2. Col 3 is the query image, while cols 4-7 illustrate the four nearest exemplar neighbors from the database, respectively. The number on top of each image is the distance to the query and the weight of contribution caculated based on that distance.

5.3 Enforcing spatial smoothness

Spatial dependencies between pixels can be used to enforce a local smoothness constraint on image segmentations. In the case of dermoscopy images, the growth pattern of PSLs results in a radiating appearance, with the result that pixels at the same distance from the center of the lesion are more likely to be statistically dependent.

This arises from the growth pattern of pigmented skin lesions. PSLs arise from melanocytes (skin pigmentation cells) and are commonly identified with two growth phases, radial and vertical. In the radial growth phase, both melanoma and benign pigmented lesions appear as plaques. Lesions increase in depth and carry a less favorable prognosis when they enter the vertical growth phase (see Fig. 33(a)). Since skin absorbs and scatters light, the appearance of melanocytes varies with depth. As a result, melanocytes are dark brown within the epidermis, tan at or near the



(a) (b) (c)

Figure 33: (a) The growth pattern of PSLs (Image courtesy of Med-Art) (b) The appearance of pigmented cells varying with depth (Image taken from [72], used with permission) (c) The radiating appearance of PSLs on the skin surface.

dermoepidermal junction, and blue-gray in the dermis (see Fig. 33(b)). This gives pigmented lesions a common radiating growth pattern seen from skin surface. That is in general, the difference in lesion appearance is more significant along the radial direction from the lesion center than any other direction, as shown in Fig. 33(c). Notice that the skin patch in the cyan square bears more resemblance to the patch in the red square than the yellow one, even though the yellow square is much closer to the cyan square.

5.3.1 Using polar radius

Many algorithms enforce explicit spatial constraints to simplify the figure/ground label assignment. Melli *et al.* [71] assume that the pixels at the four corners of an image belong to skin. Celebi *et al.* [16] discard the rectangular regions touching the image frame. Moreover, some methods such as the mean-shift algorithm implicitly enforce local neighborhood constraints on image Cartesian coordinates during pixel grouping. However, because pigmented lesions have a radiating appearance, direct embedding of Cartesian coordinates may not be optimal. For example, in Cartesian coordinates, the red square in Fig. 33(c) is farther away from the cyan square than the yellow one. Consequently, it is less likely to be grouped together with the cyan square, even though their underlying melanocytes are probably in the same growth

period. Given the fact that dermatologists tend to put the lesion near the center of image frame while acquiring dermoscopy images, we can better capture the radial growth pattern by replacing the x, y coordinates with a polar radius r measured from the image center.

We verify this by comparing the clustering residue errors using both coordinate systems. The feature space includes the coordinates of each pixel, (x, y) in Cartesian and r in polar, and its RGB values. We apply *k-means* [2] clustering on both representations of the image. After clustering, we generate a *filtered image* by substituting the RGB values of each pixel with its cluster mean. The mean per-pixel residue between each original and filtered pair is then calculated. We apply this operation on our dermoscopy dataset (*Derm*), which consists of 216 lesion images, as well as on the Berkeley natural image dataset (*BSD*) [70], which has 300 images of randomly selected natural scenes

Fig. 34 shows that for Dermoscopy images in Derm, the mean residue is reduced by 18% switching from Cartesian to polar coordinates. In contrast, the difference is much less for the natural images in BSD. Moreover, in the Derm data, all but two data points have smaller residue error under polar system, whereas for the BSD, the data points are being spread more uniformly across the equal performance line. We will describe our segmentation algorithm in the next section and demonstrate how the embedding of spatial constraints improves segmentation performance.

5.3.2 Incorporating dermoscopic spatial constraints

To incorporate such spatial constraints into our segmentation algorithm, we do the following: The pixel color vector is augmented with a normalized polar radius which is 0 at the center of the lesion and 1 at the edge. *k-means++* clustering is then used to group the 4-D feature vectors into segments. Fig. 35 illustrates the identified segments for a typical lesion image. Notice the annular rings which follow the growth

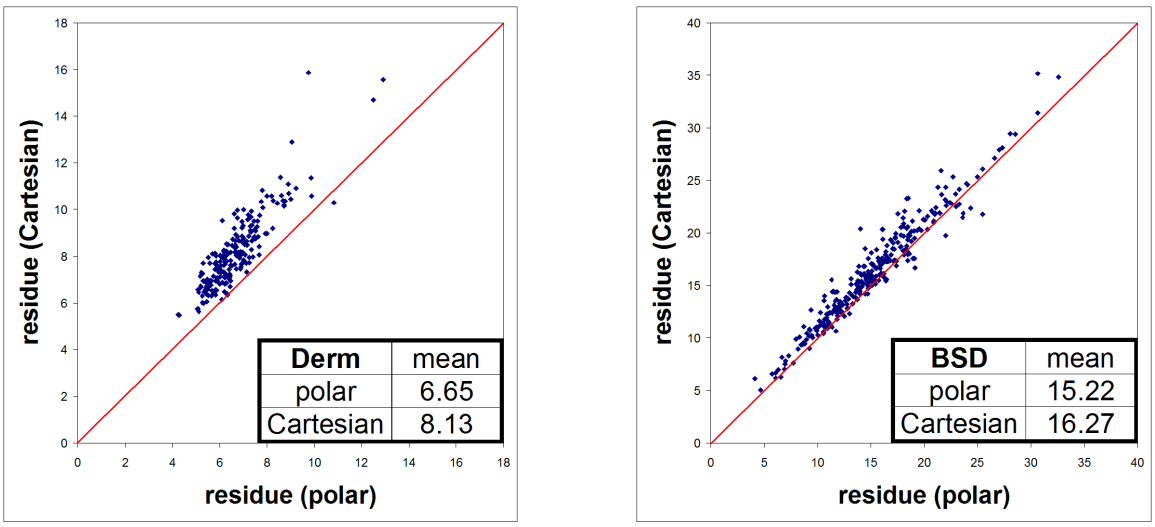


Figure 34: Mean per-pixel residue comparison between polar and Cartesian representation on the Derm dataset (left) and BSD dataset (right).

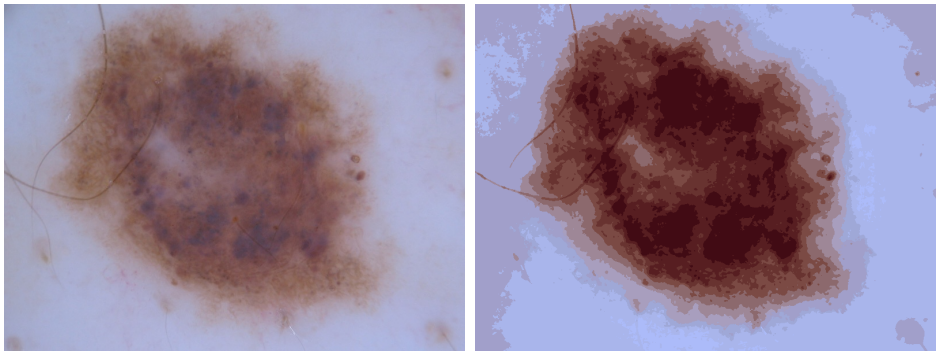


Figure 35: Dermoscopic specific radiating appearance captured by the clustering step.

pattern. We use these segments to smooth the probability estimates: Within each segment, we compute the average posterior according to Eq. 8 and take the max to obtain the label for that segment. By using a large number of segments (32 in our examples) we obtain a detailed segmentation which respects spatial continuity in a manner that is consistent with the growth patterns of PSLs. The final segmentation boundary is obtained by post-processing the mask with connected component analysis and morphological operations to remove holes.

5.4 Evaluation

We evaluated the performance of our method on three dermoscopy datasets (Table 3.3.1) with ground truth segmentations. Dataset DS1 consists of 67 images labeled by two expert dermatologists, and was provided by the authors of [107]. Dataset DS2 contains 111 images, and includes many difficult cases such as examples in the first and third rows in Fig. 36. It was labeled by our two expert dermatologist collaborators. Dataset DS3 combines DS1 and DS2 with an additional 1610 images from a variety sources, including standard dermoscopy books [72, 1]. Ground truth labels for these additional images were provided by a skilled operator.[2]

We conducted a total of five experiments in which we compared the results from our exemplar-based classifier (EBC) and its spatially-smoothed version (SEBC)[3] to three competing methods: JSEG [15], SRM [16], and SCS [107]. Using software provided by the authors,[4] we tested each method on our three datasets. For datasets D1 and D2, we performed leave-one-out cross-validation, treating each image as a query and using the remaining images as exemplars. For D3, 550 exemplars were randomly chosen and the remaining 1787 images were used for testing. Segmentations were scored using the standard XOR metric [16], which reports the area of the XOR between the prediction and groundtruth masks, normalized by the ground truth area.[5]

Fig. 37 shows the average XOR errors for all methods. Both EBC and SEBC consistently out-performed all other methods, frequently by a large margin. In the

[2]The images in D1 and D2 are cropped and scaled to 800×600 pixels. D3 includes images of various aspect ratios and resolutions, the majority of which are 4 by 3 and have a width of around 750 pixels.

[3]The parameters for our method include n, the number of histogram bins per dimension, K, the number of neighbors, and N, the number of clusters for spatial smoothing. These were tuned on a randomly-selected subset of images from D2 and fixed across all of the experiments. The settings were $n = 20$, $K = 5$, $N = 20$. We explored several color spaces (RGB chosen) and histogram distance measures, including intersection, correlation, chi-square, and Bhattacharyya (chosen).

[4]The JSEG software failed for some images in D3, and so we omitted results for that combination.

[5]Other measures such as the Dice measure or Hausdorff distance between the two lesion borders may also be used as the metric; however, we use the XOR metric because it is the standard in the lesion segmentation literature.

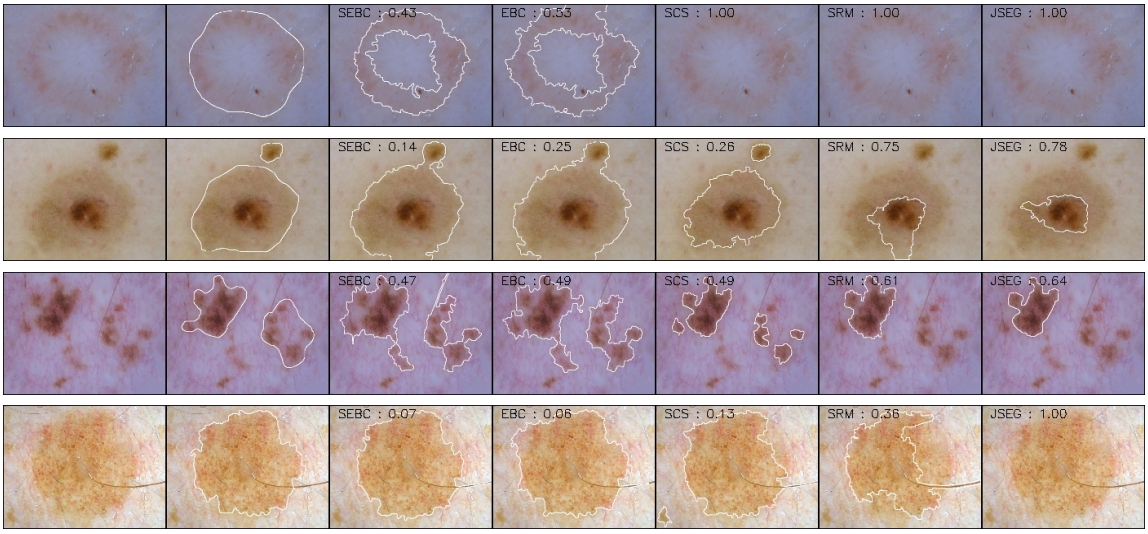


Figure 36: Segmentation results for the images of Fig. 32. Col 1 is the image, col 2 is the groundtruth. Cols 3-7 are results generated by the SEBC, EBC, SCS, SRM, and JSEG methods.

	IOE	**SEBC**	**EBC**	SCS	SRM	JSEG
DS1(67)	11.32	**13.36**	13.70	14.93	20.77	20.43
DS2(110)	13.72	**25.88**	26.76	28.77	39.50	32.81
DS3(1787)	-	**20.62**	22.23	39.58	36.77	-
Time(sec)	-	4.37	**0.45**	5.72	**0.46**	9.67

Figure 37: Percentage border error statistics using the XOR grading system. IOE stands for Intro-operator error, which are discrepancies in ground-truth segmentation between two experts.

case of datasets D1 and D2, the existence of two separate hand-segmentations allowed us to estimate the intra-operator error, which was 11.32 for D1 and 13.72 for D2.[6]

Note that on dataset D1, SEBC came within 2% of the expert intra-operator error. All experiments were performed on a Intel Pentium D processor clocked at 3.20GHz with 3.25 GB of memory. No parallel processing was enabled. The average execution time in seconds was computed for dataset D2 and is reported in the fifth column of Fig. 37. EBC and SRM were the fastest.

[6]These errors measure the difference between the segmentations produced by two sets of expert dermatologists on datasets D1 and D2, and quantify the challenges posed by these datasets even for expert clinicians. Note however that the doctors' performance is remarkably consistent when moving from D1 to the more difficult dataset D2, while the performance of all algorithms degraded considerably.

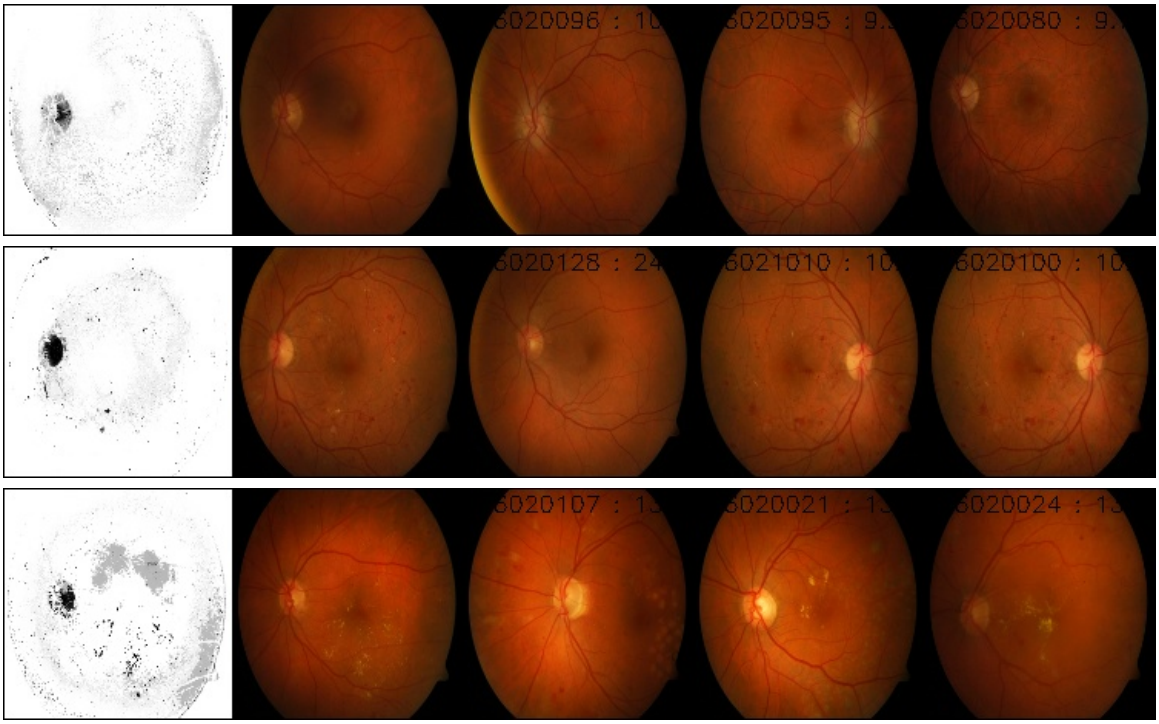


Figure 38: Nerve head segmentation in fundus images. Col 1 is segmentation (dark is high probability), 2 is query image, and 3-5 are the nearest exemplars.

Fig. 36 shows example segmentations for the images of Fig. 32, which include several difficult cases. Due to low contrast, the lesion in the first row is missed by all previous methods. Our approach is successful because it can exploit contextual information from neighbors with similar appearance (as illustrated in row 1 of Fig. 32). Similarly in row 2, the ability to retrieve neighbors with compound appearance results in improved performance. Comparing the results from SEBC and EBC (columns 2 and 3), we find that incorporating dermoscopy-specific spatial constraints improves border localization and segmentation quality.

We conducted an additional experiment to explore the applicability of the EBC method to non-dermoscopic images. Fig. 38 shows the results from segmenting the optic nerve head (a white circular spot) in fundus images of the retina. This dataset has completely different properties from dermoscopy, and yet our method yields promising results.

5.5 Related work

Existing segmentation methods can be roughly classified into the categories: edge/contour-based [31], region-merging/clustering-based [88, 16, 107], and thresholding/classification [48, 41] methods (see [17] for a detailed review). Unsupervised methods that attempt to cluster the feature data into lesion and skin distributions are unlikely to succeed due to the overlap between these classes. Our approach is most similar to supervised learning methods which train a classifier using labeled examples (e.g. the neural network approach in [48]). In fact, our technique is most related to SVM classifiers, which employ a kernel distance which is related to our d_k measure. The primary advantage of our approach in comparison to SVM and other supervised methods is the ability to adjust the set of exemplars on-the-fly, effectively tuning the neighborhood definition for a query image. We are employing SEBC within a larger system for computer-aided diagnosis of dermoscopy images.

Our system allows clinicians to interactively query a database of annotated dermoscopy images. By choosing relevant examples the operator can easily refine the segmentation produced by SEBC. Such flexibility would be difficult to obtain using an offline-trained supervised learning method.

5.6 Conclusion

We have described our novel lesion segmentation algorithm for dermoscopy images that leverages a set of exemplars annotated by experts and incorporates dermoscopy-specific spatial smoothing. Our method consistently outperforms three existing approaches on a large hand-segmented dataset. It is computationally efficient and simple to implement, and potentially effective in other problem domains.

CHAPTER VI

A GENERALIZED REPRESENTATION OF DERMOSCOPIC FEATURES FOR DETECTION AND MATCHING

6.1 Introduction

Dermoscopic features provide important visual clues to clinicians for diagnosing Pigmented Skin Lesions (PSLs). Automated detection, classification, and matching of these visual features is an important task in the computer-aided diagnosis of PSLs, but the large number of these features and the subtle distinctions among them make this goal difficult to achieve. In this chapter, we present a general representation for dermoscopic features using the bag-of-visual-words (BoVW) framework. We demonstrate that this representation can characterize common dermoscopic features. Moreover, we show that it also enables an efficient region-of-interest based search method that matches visually similar features in a large annotated database of examples. Our experimental results indicate that this general representation can be utilized for computer-aided diagnosis of melanoma and PSLs in general.[1]

There are two basic requirements for a CAD system for melanoma screening and diagnosis: a *diagnostic scoring capability* which can analyze a dermoscopic image and assess the risk of melanoma, and a *retrieval capability* which can allow a clinician to retrieve relevant examples of dermoscopic images from a database of examples with associated diagnoses. The retrieval capability makes it possible to quickly examine additional annotated examples in refining the assessment of risk. Both risk-assessment

[1]Part of the work described in this chapter has been published in [109].

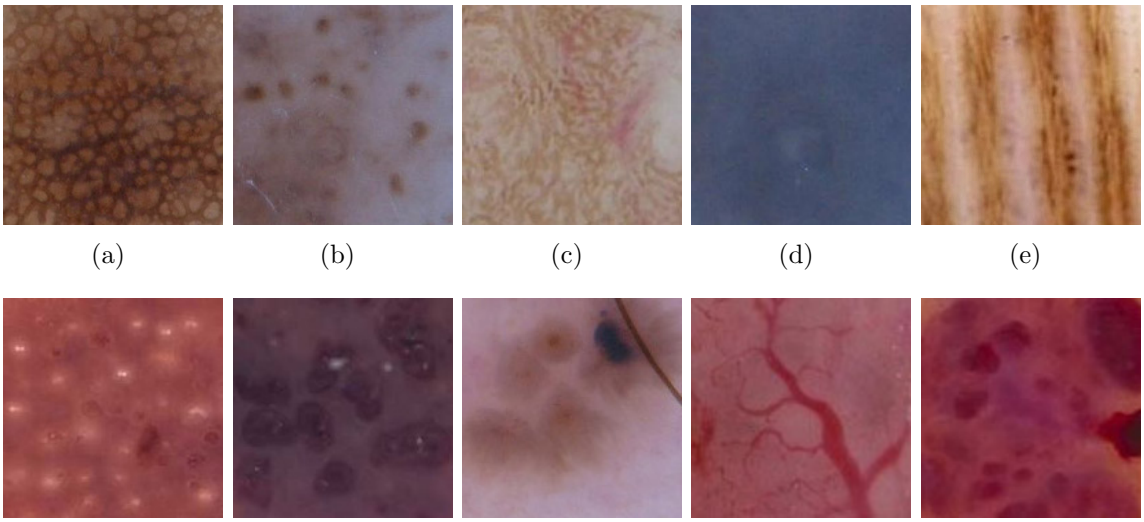


Figure 39: Examples of dermoscopic features: a. pigment network, b. brown dots/globules, c. streaks, d. homogeneous blue pigmentation, e. parallel pattern (on palms and soles), f. milia-like cysts, g. comedo-like openings (typically seen in seborrheic keratoses), h. maple leaf-like areas, i. arborizing vessels (typically seen in basal cell carcinoma), and j. red lacunas (typically seen in vascular lesions).

and retrieval in turn require effective methods for representing and analyzing dermatological features (see Fig. 39 for examples).

This chapter addresses the characterization of dermatological features to support CAD for melanoma. We present a novel computational model for these features, which is based on a *visual words representation*. Our model expresses complex dermoscopic features as the combination of underlying basis patterns of pigmentation. Our experimental results indicate that this general representation can be utilized to detect a diverse set of dermatological features in pigmented skin lesions. Furthermore, in the next few chapters, we will demonstrate that this visual words model can address the two main requirements for CAD: the implementation of a standard dermatological pattern scoring function, known as *ABCD Analysis* (detailed in Section 8.1.1), for diagnostic scoring; and the ability to perform *Region of Interest (ROI) search* (described in Section 7.2.2) in retrieving relevant examples containing specific

dermatological features of interest from a database.

6.2 Related work

Previous works on automated diagnosis of melanoma [53, 13, 8, 39] typically adopt a supervised classification approach in which a dermoscopy image is analyzed by a classifier which outputs either a binary decision or a probability score for the presence of melanoma. A representative example is the work of Iyatomi et. al. [53], who developed an Internet-based diagnosis system which uses a neural network classifier. A disadvantage of such "black-box" classifiers is the difficulty of post-hoc interpretation and analysis of their decisions. In a CAD setting, risk assessment should ideally be based on a clinically-relevant scoring system, not the weights in a neural network or support vectors in an SVM.

The identification and characterization of dermoscopic features is a key task in the analysis of dermoscopic images. These features play an important role in diagnostic scoring systems such as ABCD, and they are the primary cue used by dermatologists in identifying related image examples. Previous work in automatic analysis of dermoscopic features has been limited in scope, with researchers typically addressing only a few types of feature in isolation. For example, Betta *et al.* [6] developed techniques to detect and analyze atypical pigmented network and vascular patterns. Grana *et al.* [42] detected curvilinear features to characterize network patterns. Tanaka *et al.* [93] applied texture analysis techniques to classify three common global patterns typical to melanocytic lesions. Iyatomi *et al.* [52] developed a procedure to classify parallel furrow and ridge patterns that are characteristic to melanocytic acral lesions.

These previous works suffer from two broad limitations. First, they use binary classifiers to detect the presence of individual dermatological features. In contrast, retrieval systems require a similarity score between examples that can be used for ranking. Second, these methods adopt a global representational approach, in which

information across the entire dermoscopic image is encoded in the feature vector. In practice, multiple diverse dermatological features can often co-exist in a single image. This has two consequences: the complexity of the concept class is greatly increased (the classifier must learn about conjunctions of features) and it suggests that similarity scores between images at the global level may not be very useful (it's not clear how much each feature should contribute to the comparison). In contrast, our approach based on local region analysis has two benefits: the occurrence of multiple, spatially-localized features can be identified within an input image, and a user can select a desired ROI in order to search for examples which contain a particular type of feature.

6.3 Approach

There are over one hundred dermoscopic features commonly seen in clinical studies, but upon close examination, many of these features share low level image characteristics such as ridges, blobs, and streaks. It is the difference in the concentration and spatial configuration of these characteristics that gives rise to the diverse properties of dermoscopic features. This suggests that individual dermoscopic features could be described using a vocabulary of local spatial features corresponding to primitive elements such as ridges and blobs. In the computer vision community, this is known as a Bag of Visual Words (BoVW) representation, and it has been shown to be effective for visual object categorization.

Fig. 40 illustrates our approach to the construction of a BoVW feature descriptor and its use in diagnosis and retrieval. The starting point is the identification of local interest points (local spatial regions in the dermoscopy image) which are salient to the overall pattern. Methods such as SIFT [68] and SURF [4] have been developed to detect salient local points in camera images. Here, we propose *DIP* (short for Dermoscopic Interest Point): a feature detector specifically designed to extract local

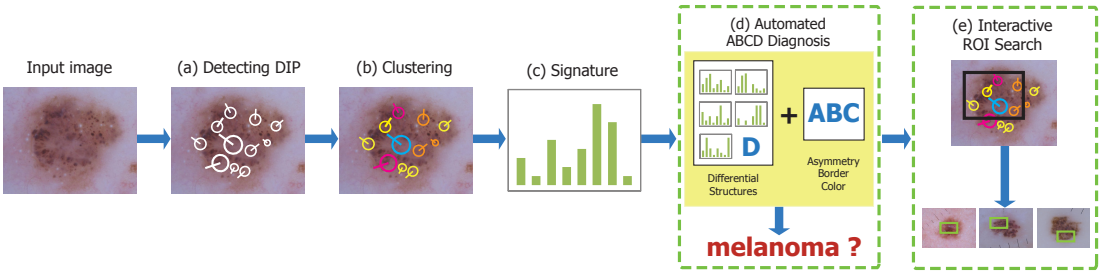


Figure 40: Block diagram illustrating the construction and use of dermoscopic feature descriptors.

intensity structures in dermoscopic images and a descriptor for constructing a general visual vocabulary for dermoscopic features. To the best of our knowledge, this is the first time the concept of interest points for dermoscopic features is introduced. Like SIFT and SURF, DIP descriptor is scale and rotation invariant to handle features undergo geometric deformations (restricted to a plane due to how dermoscopy images are acquired), and it models photometric deformations using a linear model with a scale factor and offset. However, DIP differs from previous local interest point detectors and descriptors in two places. First, DIP detector not only selects distinctive locations from corners, blobs, and junctions, but it also include regions where strong curvilinear features reside. This is due to the prevalent presence of curvilinear features in dermoscopy images. Second, DIP stores color information in its descriptor since color cues are important for discriminating many dermoscopic features in pigmented skin lesions. Given Dermoscopic Interest Points *DIP*s from a set of training images, we can cluster their descriptors to obtain the BoVW model. For any spatial region in an input image, a signature can be constructed from the distribution of visual words. This can be used to detect known features (in the case of ABCD diagnosis) and match similar patterns (in the case of retrieval). We now describe each step in more detail.

6.3.1 Detector

Given a dermoscopy image, we select interest points at distinctive locations inside the lesion such as corners, blobs, junctions, and streaks. We want our interest point detector to be repeatable, i.e. it should reliably find the same interest points under different viewing conditions. In general, vision interest point detectors such as SIFT [68] and SURF [4] focus on corners and blobs because these image characteristics are robust to various image transformations. This applies to dermoscopy images as well; therefore, we start with detecting corners and blob structures. In addition, many dermoscopic features, such as the pigmented networks shown in Fig. 39(a), have strong curvilinear components. In order to capture those features, we augment our detector with a component that specifically locates curvilinear structures.

6.3.1.1 Corners and blobs

We adopt the fast-Hessian detector proposed in Bay, *et al.* [4] to locate corner and blob structures. Given a dermoscopy image J, we first convert its pixels from RGB to $L^*a^*b^*$ values since the $CIELAB$ space is more perceptually uniform, and the L channel roughly captures the luminous component of the image. We use the determinant of the Hessian matrix for selecting both the location and scale of each interest point. Given a point $\vec{x} = (x, y)$ in the intensity channel L, the Hessian matrix $\mathcal{H}(\vec{x}, \sigma)$ in $\vec{x}$ at scale σ is defined as follows

$$\mathcal{H}(\vec{x}, \sigma) = \begin{bmatrix} L_{xx}(\vec{x}, \sigma) & L_{xy}(\vec{x}, \sigma) \\ L_{xy}(\vec{x}, \sigma) & L_{yy}(\vec{x}, \sigma) \end{bmatrix}, \tag{9}$$

where $L_{xx}(\vec{x}, \sigma)$ is the image I convoluted with the Gaussian second order derivative $\frac{\delta^2}{\delta x^2} g(\sigma)$, and likewise for L_{xy} and L_{yy}. The detector approximates second order Gaussian derivatives with box filters (Fig. 41), denoted as D_{xx}, D_{yy}, and D_{xy}, which can be evaluated efficiently using integral images. We use 9×9 box filters at the lowest scale $\sigma_0 = 1.2$. The filter response is computed as $det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2$, where

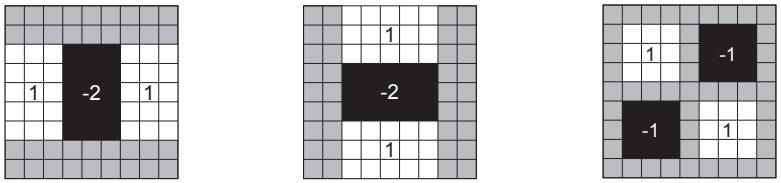


Figure 41: Gaussian second order partial derivatives in x-, y-, and xy-direction approximated using box filters. Grey is zero.

0.9 is used to balance the relative weights from the approximations. The responses are normalized with respect to the mask size to guarantee a constant Frobenius norm for any filter size. Scale space computation is also made easy by the use of box filters and integral images. We build each image layer, referred as $s = \sigma$, by filtering the original image with gradually increasing masks (e.g. 9×9, 15×15, 21×21) instead of iteratively down-sampling the image. As a result, the same integral image can be reused and the speed is exactly the same for any filter size. The Gaussian derivatives scale along with the filters. For example, the 27×27 filter corresponds to $\sigma = 3 \times \sigma_0 = 3.6 = s$. After applying non-maximum suppression in a $3 \times 3 \times 3$ neighborhood of the image layers, we interpolate the Hessian determinant in both scale and image space. We then threshold the interpolated maximum responses to exact candidate sites for corner and blob structures.

6.3.1.2 Curvilinear structures

To locate strong curvilinear components, we apply Steger's line point detection algorithm [91] to the intensity channel L. Line points are points in an intensity image where the first directional derivative in the direction of the line vanishes, and the second directional derivative has a large absolute value. A point $\vec{x} = (x, y)$ is a line point

if it satisfies $(tn_x, tn_y) \in [-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$, where (n_x, n_y) is the normalized eigenvector[2] that corresponds to the maximum absolute eigenvalue of the local Hessian matrix $\mathcal{H}(\vec{x}, \sigma)$, and t is evaluated as follows:

$$t = -\frac{L_x(\sigma)n_x + L_y(\sigma)n_y}{L_{xx}(\sigma)n_x^2 + 2L_{xy}(\sigma)n_x n_y + L_{yy}(\sigma)n_y^2}, \tag{10}$$

where L_x, L_y, L_{xx}, L_{xy}, L_{yy} are partial derivatives of the image estimated by convolving the image with discrete two-dimensional Gaussian partial derivative kernels. The standard deviation σ of these kernels is directly tied to the expected line width. Therefore, we apply the line detection algorithm at multiple scales (with different σ's) to extract line segments within a certain width range. The saliency of a line point (x, y), i.e. the absolute value of the second directional derivative along (n_x, n_y), is proportional to its intensity. We find local maximum filter responses in both scale and image space to locate strong curvilinear structures.

6.3.2 Descriptor

The distinctive power of state-of-the-art interest point descriptors relies on a combination of approximately spatially localized information and the distribution of gradient-related features. Relative strengths and orientations are often used instead of absolute ones to reduce the effect of photometric changes. The proposed dermoscopic feature descriptor is based on similar properties, with the addition of linear and color components. We first identify a reproducible orientation based on local statistics calculated from a circular region around each interest point. We then construct a square region aligned to this orientation and extract a feature vector from it.

6.3.2.1 Orientation

Once the location of an interest point is determined, we identify a reproducible orientation in order to achieve rotation invariance. We first compute the Haar-wavelet

[2] (n_x, n_y) points in the direction perpendicular to the line direction at point (x, y).

responses in both x and y direction in a circular neighborhood of radius $6s$ around the interest point, with s being the scale of the interest point. The sampling step and the scale at which we compute wavelet responses are chosen as s, so the wavelets are large at high scales. Using the same integral images introduced in the last section, only six operations are needed to compute the wavelet response in both directions at any scale. The resulting responses, after being weighted by a Gaussian, are represented as vectors in a $2D$ space with the horizontal and vertical response strength as coordinate axes. The dominant orientation is estimated by summing up all the respective horizontal and vertical responses in a sliding window covering an angle of $\frac{\Pi}{3}$. The two summed responses then yield a new vector, and the orientation of the longest such vector is chosen as the orientation of the interest point.

6.3.2.2 Descriptor components

Once a reproducible orientation is identified, we construct a square region oriented along this direction, and centered around the interest point. The size of the square is $20s$ and it defines the context of our descriptor. After dividing the context region uniformly into 4×4 sub-regions, we compute four simple features at 5×5 regularly spaced sample points for each sub-region. We use d_x, d_y to denote the Haar wavelet responses in horizontal and vertical directions (filter size $2s$). We also extract the absolute values of the responses, denoted by $|d_x|$ and $|d_y|$, to register the polarity of the intensity changes. After summing up these individual measurements within each sub-region, we obtain a four-dimensional descriptor vector v with underlying intensity structure $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Therefore, for each interest point, the resulting descriptor vector is of length 64 for all 4×4 sub-regions. The wavelet responses are already invariant to illumination changes, and we can further achieve contrast invariance by normalizing the descriptor into a unit vector.

In addition to intensity statistics, we also want to include color information, as it often plays a vital role in the diagnosis of PSLs; the appearance of pigmented skin

cells (*melanocytes*) varies with depth and results in different dermoscopic features. For instance, black globules and brown globules may have very similar gradient profiles, but they correspond to different pathologies and should be distinguished. To accomplish that, we include color statistics in our descriptor using the W invariant, which is derived from the opponent color space as $W_1 = \frac{O_1}{O_3}$ and $W_2 = \frac{O_2}{O_3}$. Channel O_1 and O_2 store the color information, and the division by intensity channel O_3 makes this statistic intensity-invariant. To form a color-aware descriptor, the intensity statistics are simply calculated for both W channels and concatenated. This approach has also been shown to be a reliable color descriptor for general recognition tasks [96].

6.3.3 Bag of visual words descriptor

Dermoscopic interest points enable us to represent dermoscopic features using a *bag of visual words (bovw)* model. Given a set of training images with lesion segmentations, we extract local DIP descriptors within the lesion boundary for each image. We vector-quantize these DIP descriptors using a K-entry visual word codebook, which is obtained from running K-means clustering algorithm on a randomly sampled subset of the descriptors. After quantization, each DIP is associated with a discrete visual word index $c_j \in \{1, \ldots, K\}$. Figure 42(a)-(e) show example patches of the five most frequently occurring words in our DIP codebook. Each of these columns lists the patches that are closest to their respective cluster centers, and they appear to correspond to specific image features. With the visual codebook, we can now represent images or regions within images (Fig. 42(f)-(i) top row) with their cluster histogram by counting how many DIP of each cluster index occur. The corresponding cluster histograms of the example patches are shown in the middle row of Fig. 42(f)-(i).

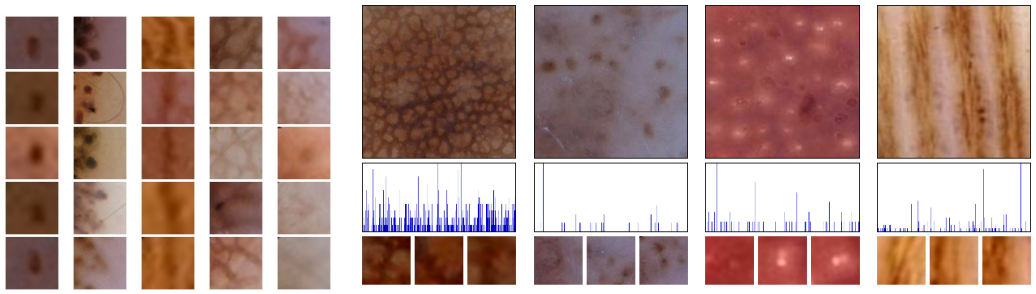


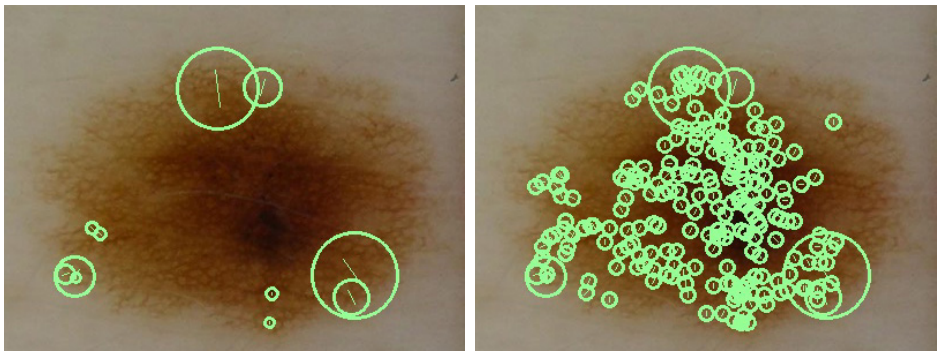
Figure 42: (a-e) 5 most frequently occurring visual word. They correspond to (a) brown blob, (b) globules on the boundary, (c) brown ridge, (d) network, (e) light indistinguishable texture pattern. Notice that descriptors are rotation invariant but these patches are intentionally aligned based on their descriptor orientations. (f-i) Typical DFs (top), corresponding signatures in the *bovw* representation (middle), and the three most occurring visual words in the patch. From left to right, the DFs are (f) discrete network, (g) globules, (c) milia-like cysts, and (d) parallel furrow patterns.

6.4 Evaluation

6.4.1 Detector and descriptor

We evaluate our representations on DF1 (Table 3.3.1): a dataset of 150 dermoscopy images containing common PSLs. At least one dermoscopic feature is present within each lesion boundary. The features are outlined and annotated by our collaborating dermatologists. For the detectors and descriptors used in comparison, SURF is based on the original implementation of the authors, and SIFT is from a relatively efficient implementation [97] based on the original publication. We first compare how sensitive these detectors are to dermoscopic features. We then check their repeatability on dermoscopy images undergoing common transformations.

For each image, all the detector responses within the lesion boundary are retrieved. Those points that land inside dermatologists' manual feature outlines are considered relevant. The starting threshold for each detector is set to a level low enough to generate a large number of responses. These responses at the lowest threshold are used as the relevant feature set for each detector. As we gradually increase the threshold,



(a) (b)

Figure 43: Interest points detected by the SURF (left) and DIP (right) detectors.

fewer responses are produced. We plot the precision-recall graph in Figure 44(a). Our experiment shows that the DIP detector consistently outperformed both the SURF and SIFT detector on dermoscopy images. Figure 43 shows an illustrative comparison between SURF and DIP detection results. The lesion in the images exhibits a common dermoscopy feature called pigmented networks. There are only a few corners and blobs strong enough to trigger SURF detector responses, and the majority of the interesting feature region is overlooked.[3] Under similar settings, the DIP detector captures more interest points on the same lesion. This is true in general, and according to the precision-recall graph, a higher percentage of these DIP responses are relevant to dermoscopic features.

An effective interest point detector and descriptor should reliably find the same interest points under different viewing conditions. For example, reliable feature correspondences are needed in applications such as lesion registration and change detection for longitudinal studies. To demonstrate the repeatability of the DIP detector and descriptor, we set up the following experiment. We perform a set of scale, rotation,

[3]Similar results are observed when the SIFT detector is used. The image is not shown due to space limitation.

and lighting change operations on each image. The detector responses at each location before and after the change are matched; any inconsistency indicates a miss detection. When both responses are present (within a three pixel radius), we calculate the Euclidean distance between their descriptor vectors. If this distance is larger than a preset threshold, we consider the pair as a miss match. Figure 44(b), 44(c), and 44(d) shows the repeatability comparison results for decreasing light, scale, and rotation change, respectively. Although under similar settings, the DIP detector often extracts more interest points than SIFT and SURF, and the additional color components in its descriptor can increase matching difficulties, its repeatability, as evidenced in the graph, is on par with SURF, and slightly superior to SIFT. All the experiments were performed on a standard Windows PC (Pentium D, 3.20 GHz).

6.4.2 Bag of visual words descriptor

The effectiveness of the proposed BoVW representation for dermoscopic features will become evident as the feature descriptors are used for pigmented skin lesion diagnosis during the evaluation of the DermFind system (Section 8.1).

6.5 Conclusion

This chapter addresses the characterization of dermatological features to support CAD for pigmented skin lesion diagnosis. We present a novel computational model for these features, which is based on a *visual words representation*. Our model expresses complex dermoscopic features as the combination of underlying basis patterns of pigmentation, and our experimental results indicate that this general representation can be utilized to detect a diverse set of dermatological features in pigmented skin lesions. In the following chapters, we will demonstrate that this visual words model can address the two main requirements for CAD: the ability to perform interactive *Region of Interest (ROI) search* (Section 7.2.2) in retrieving relevant examples

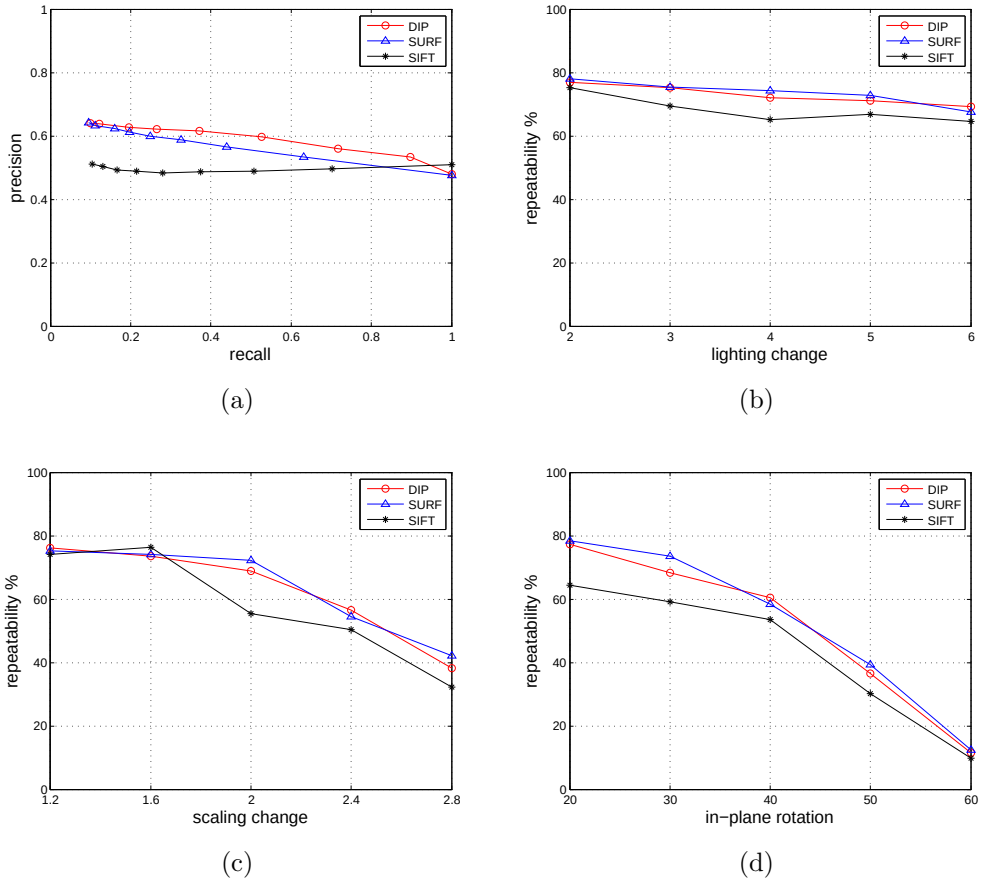


Figure 44: (a), (b), (c), and (d) are from left to right and top to bottom. The precision-recall graph (a) demonstrates that the DIP detector is more sensitive to dermoscopic features. (b)-(d) show that DIP achieves a comparable level of invariance to lighting change, scale change, and in-plane rotation when measured against SIFT and SURF.

containing specific dermatological features of interest from a database, and the implementation of a standard dermatological pattern scoring function, known as *ABCD Analysis* (Section 8.1.1), for diagnostic scoring.

CHAPTER VII

INTERACTIVE IMAGE AND REGION-OF-INTEREST RETRIEVAL

The key strength of DermFind, which differentiates it from many conventional computer-aided diagnosis platform, is the capability of content-based image retrieval at its core. This not only allows dermatologists to browse through retrieved exemplars to obtain an intuitive understanding of the diagnosis, but it also enables them to provide feedback and refine analysis through manipulation of retrieved exemplars. Two components are indispensable for effective search and retrieval at interactive speed: one is the definition of image similarity criteria, particularly when multiple criteria are conjointly considered, the other is the efficiency of the search and comparison operation. In this chapter, we will focus on techniques that enable meaningful, expedient retrieval in DermFind.

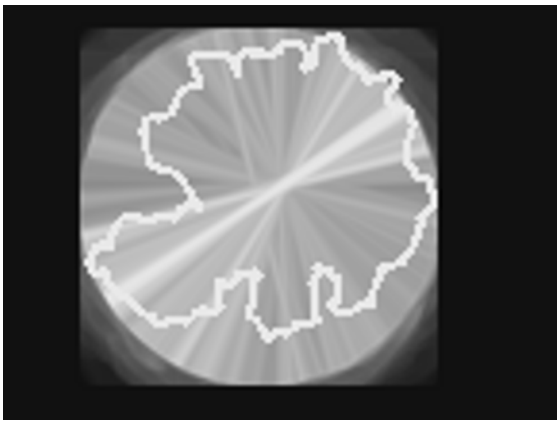
7.1 Search criteria

Individual features, compound features, dermatological meaningful criteria (ABCD analysis), and region of interests are the retrieval criteria employed in DermFind. This section provides a detailed description for all of them.

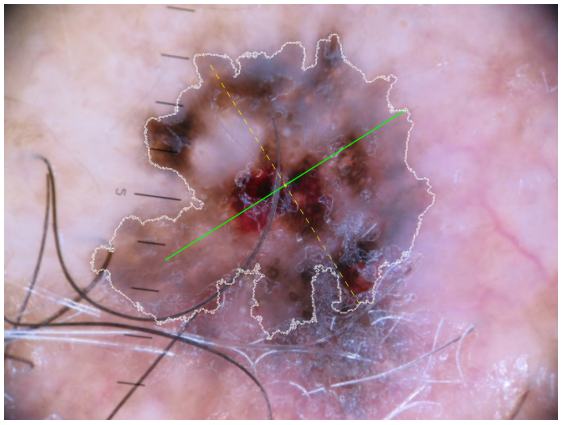
7.1.1 Individual feature as criteria

A key strength of DermFind is a state-of-the-art feature analysis module, which provides the feature attributes described in detail below. The development of a set of features and mechanisms for their combination mirror the analysis criteria of a skilled dermatologist. We developed our feature model by working closely with domain experts and capturing their insights.These features can be combined to obtain

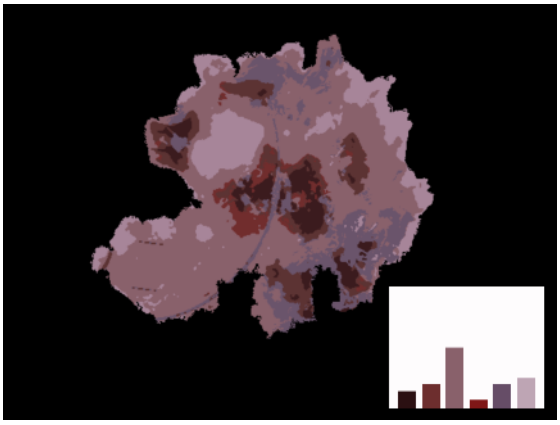
(a) Original dermoscopy image



(b) Planar Reflective Symmetry Transformation of the lesion



(c) Lesion with symmetry axes and border overlayed



(d) Six color quantization of the lesion

Figure 45: Individual feature search criteria.

the retrieval criteria described in Section. 7.1.2 and Section. 7.1.3, but they can also be used as individual search criteria. We now describe these feature attributes and how they are computed in more detail.

Asymmetry: We extract the major and minor symmetry axes from the shape delineated by the lesion border using the Planar Reflective Symmetry Transformation (PRST) [79]. Figure 45(b) shows the result of PRST applied to the lesion shown input image (a).[1] The brightness indicates the degree of symmetry, and the brightest of

[1]PRST is actually applied to the lesion border, which is shown overlayed on top of the lesion in Fig. 45(c).

the axes (about 45 °) is the major symmetry axis (solid green line in (c)). The minor axis (dashed yellow line in (c)) is chosen as the the line that is perpendicular to the major axis and also passes through geometric center of the object. The next step is to sample the pixels from the two halves of the lesion with respect to each symmetry axis, and compute the Earth Mover's distance [84] to measure the asymmetry of the color distribution within the lesion. This process gives us two asymmetry values corresponding to the major and minor symmetry axes, respectively. Thresholding on these values results in an asymmetry score $A \in \{0, 1, 2\}$ for the lesion, where $A = 0$ indicates that the lesion is symmetric, 1 that the lesion is only asymmetric with respect to one of the axes, and 2 that the lesion is asymmetric with respect to both.

Border: This feature measures lesion border irregularity. The lesion area is divided into eight radial wedges, dissecting along the major symmetry axis. The resulting eight border sections are examined separately by measuring the average gradient magnitude inside each region surrounding the lesion boundary. These values are thresholded to give the border score $B \in \{0, 1, \cdots, 8\}$, where a score of 1 indicates that the lesion displays abruptness in border-skin transitions in one of the border sections, and so on.

Color: Dermatologists identify 6 specific color distributions (white, red, light brown, dark brown, blue-gray, and black) whose presence inside the lesion area is "alarming" (i.e. warrants further analysis). Using a training set with human annotations and the EM algorithm, we fit a Gaussian model to each of these special color clusters. The resulting color analysis process is illustrated in Fig. 45. Each pixel in an input image (a) is mapped to one of the six colors (resulting in the image (c)), yielding a histogram ((c) lower right corner). The color score $C \in \{1 - 6\}$ is the number of bins in the histogram which exceed a pre-defined threshold.

Radiating color: Histogram of lesion color is a useful feature, but it can be improved by exploiting the natural radiating growth pattern of PSLs. We design a color histogram feature based on the observation in [107] where lesions in dermoscopy images have a radiating appearance. We divide the lesion region into three concentric rings with equal thickness. A color histogram is calculated separately for each ring, and the three histogram vectors are concatenated to form a feature vector.

Texture-shape descriptors: There are over a hundred dermoscopic structures in the medical literature, and many are symptomatic. We provide a variety of texture descriptors to capture these structures. In addition to the standard GLCM [43] texture descriptor, we employ more recent ones such as BGLAM [80] and PACT [102]. PACT is based on a PCA decomposition of the histogram of the census transform (CT) of the image. The CT histogram encodes the global structure of the lesion texture. We extract the 40 top eigenvectors to form a texture-shape code book. All CT vectors are projected into this dimension 40 feature space to form a PACT feature vector. To capture global spatial variation, we construct the spatial PACT (sPACT) by building a pyramid from the original image and concatenate PACT vectors from the image pyramid.

7.1.2 The ABCD criteria

There are several standard methods in dermatology for assessing abnormality in pigmented skin lesions such as the ABCD rule, 7-point checklist, and Menzie's method. These methods provide a standardized checklist for assessing abnormality in color, shape (symmetry), texture, and structural characteristics of the lesion [1]. DermFind adapts the ABCD rule as a retrieval criteria because it is a standard pattern analysis scoring system used by clinicians to identify melanocytic lesions[2] which are at risk for melanoma. It uses four criteria: **A**symmetry, **B**order abruptness, and **C**olor,

[2]Dermoscopic features in the pigmented skin lesion are identified to determine whether the lesion is melanocytic or non-melanocytic.

and **D**ifferential structures (which are specific categories of dermoscopic features). The **A**symmetry, **B**order abruptness, and **C**olor scores are discussed in detail in Section 7.1.1. The **D**ifferential structure score is determined by detecting the presence of any of the five structural dermoscopic features: pigment network, structureless or homogeneous areas, streaks, dots, and globules. Since any lesion must have at least one of the structural features, D is a number between 1 to 5. The Total Dermoscopy Score is computed as $TDS = A \times 1.3 + B \times 0.1 + C \times 0.5 + D \times 0.5$, and a lesion with $TDS > 5.45$ is marked as melanoma according to the ABCD rule. In order to efficiently compute the differential structure score on a large image database, we train SVM classifiers to detect whether an lesion contains the five dermoscopic features. We adapt Efficient Subwindow Search [61] (described in more detail in Sec. 7.2.2) to efficiently localize dermoscopic features using the BoVW representation.

7.1.3 Compound criteria using rankings

Although search based on a measure such as the ABCD rule and TDS score is desirable as a baseline, in actual practice physicians check a number of cues simultaneously during the examination of a skin lesion. Therefore it is important to enable compound search based on multiple features. However, a simple linear sum of scalar scores (as in TDS) is not very effective and does not extend to feature vectors such as color histogram and PACT that have hundreds of dimensions. We therefore propose a similarity measure based on ranking. Unlike distance measures, ranks have a consistent meaning in any feature space. Using ranking penalizes elements that are mediocre in all feature spaces and promotes those that stand out in a few particular feature dimensions. When the selected feature dimensions are not strongly correlated, the closest neighbors determined by individual rankings are more consistent. Figure 46 illustrates the process of a search using compound criteria with ranking. The query image is displayed on the left. Right next to it is the interface displaying the retrieval

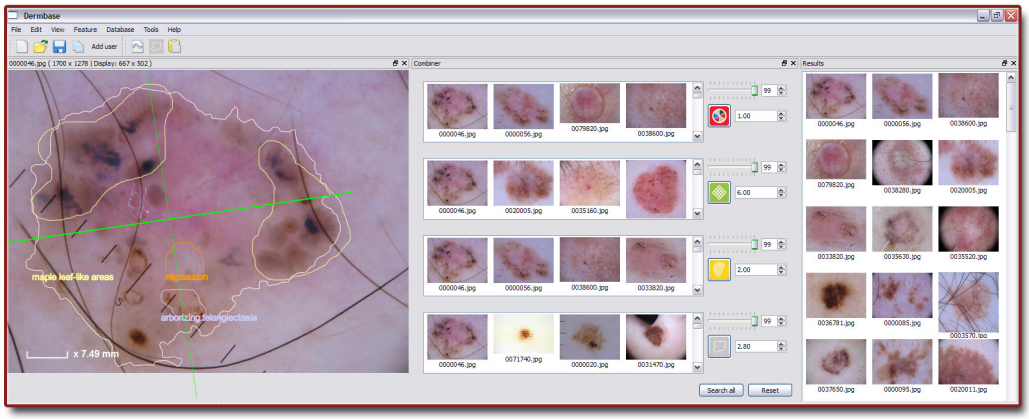


Figure 46: Search using compound criteria with rankings

results based on each individual feature criteria (from top to bottom): presence of six colors, texture similarity using sPACT analysis, layered color descriptor (radiating color), and lesion shape represented by the Hu moment vector. The right panel shows the final retrieval results after merging results from individual criteria search using their rankings.

7.1.4 Region of interest as a search criteria

The previous search criteria define similarity over the entire image, and many search techniques have been developed to efficiently locate similar images [105, 40]. However, sometimes the user is interested in just part of the image. One important application in content-based image retrieval is the problem of region of interest retrieval, (a.k.a. image part retrieval) i.e. to retrieve images from a database based on queries that only have to match a part of the query image often delineated as the region of interest (ROI). For instance, consider the following scenario in the DermFind context. A user is interested in a particular region of the lesion which contains several diagnostically significant dermoscopic features such as redial streaming, discrete pigmentation, and regression structure. To determine if the lesion is benign or malignant, the user delineates the region of interest and asks DermFind to find images that contain regions with similar appearance. Very often, one does not care about the size nor shape of

the region, but is only interested what is inside the region. The search technique that handles region of interest search criteria will be discussed in Section 7.2.2.

7.2 Search techniques

This section focuses on search techniques that enable meaningful, expedient retrieval in DermFind using the search criteria defined in the last section.

7.2.1 Search using DermTree

The power of OpenDiamond can be augmented by hierarchical data structures to improve scalability on individual nodes. We observe that there are intrinsic structures among dermoscopy images, that rather than isolated cases, they are inter-related through sharing common features, histopathology results, etc. This leads us to organize the dermoscopy images into multiple tree structures, DermTrees, to speed up similarity comparisons.

7.2.1.1 Constructing DermTree

In high dimensional space, common acceleration structures such as kd-tree or ANN are shown to be ineffective. Therefore, we propose a new approximation acceleration structure, which uses affinity propagation [38] to form a tree-like hierarchy in a bottom-up fashion. Affinity propagation identifies elements in the original dataset as cluster centers (a.k.a. exemplars). We use these exemplars to form new clusters and apply affinity propagation recursively we the remaining elements are less than a predefined threshold. An visualization of DermTree based on the "radiating color" criteria (Section 7.1.1) is shown in Figure 47.

7.2.1.2 Searching DermTree

For each user-specified similarity measure, we organize the dataset into a hierarchical tree-like representation, where children of a common parent are more similar according to the measure. Image search along the DermTree is conducted as follows: a new

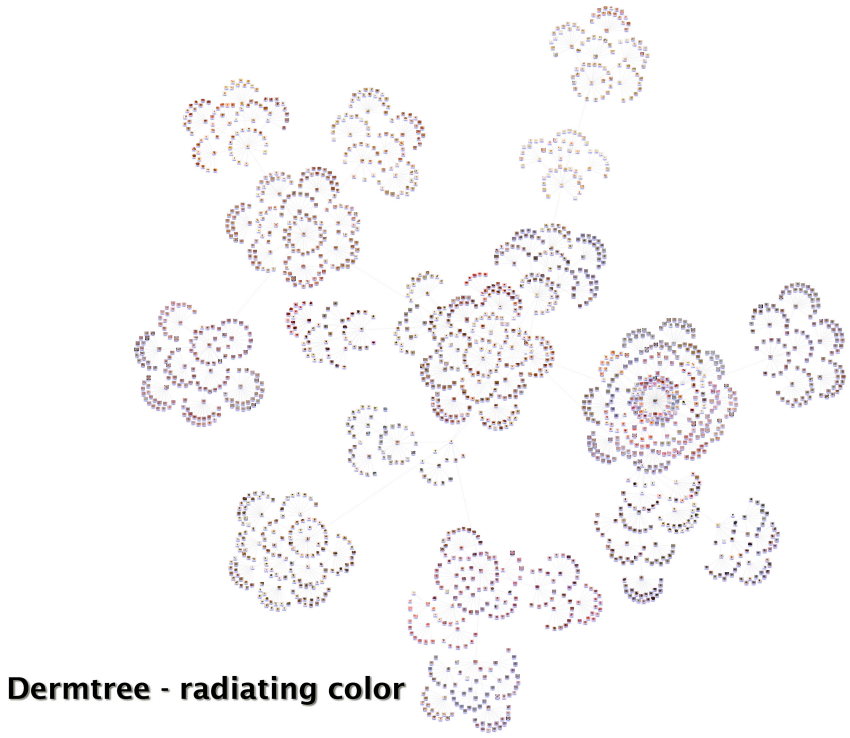


Figure 47: DermTree constructed using the "radiating color" criteria

image element is first positioned at the root. After comparing it with each child of the root, it then traverses the subtree rooted at the best matching element. All of the comparison distances are recorded. This operation is performed recursively until the element reaches a leaf. At that point, the best K matches are identified as the K nearest neighbors.

By descending through the DermTree structure, the number of comparisons are reduced from linear to logarithmic. The trade-off is that the K nearest neighbors identified in this way can be worse than they would be under brute-force search. However, in DermFind, the emphasis is on expediency and the relevance of returned neighbors (precision) rather than how big a fraction of all relevant cases we actually retrieve (recall).

7.2.2 Efficient region of interest retrieval

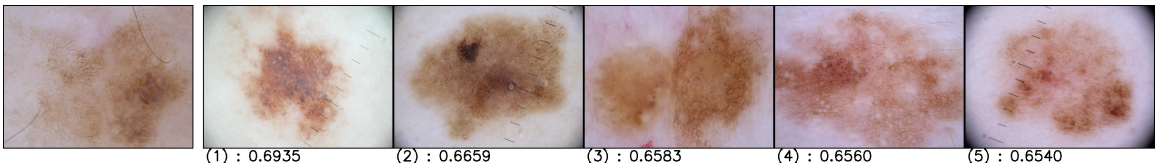
We have developed a method for dermoscopic image retrieval based on a user-defined region of interest (ROI). Such capability can enable clinicians to identify a dermoscopic feature of interest and retrieve images with similar features from a large annotated database, where the annotations can help to refine the risk assessment. A naive approach of exhaustively matching the query ROI against all possible ROIs in all images is clearly not practical even for moderate sized database. As a result, we first limit ourselves on the shape of ROIs to be boxes so that the comparison can performed in a sliding window fashion. Notice that it is still impractical to do exhaustive comparisons due to the exponential number of possible boxes within an image. However, when the *bovw* representation is used, Lampert et al. showed that it is possible to do a complete nearest-neighbor comparison between the query and all boxes in all database images using a branch-and-bound strategy in sub-linear time [61]. To enable interactive search and retrieval of ROI containing potentially interesting dermoscopic features, we adopt the this Efficient Subwindow Search *ESS* framework.

To start, we first compute a visual word representation for a query image I using Dermoscopic Interest Point (Chapter 6). This allows us to represent arbitrary image regions R by their bag of visual word *bovw* histogram h^R. ESS uses a geometric branch-and-bound optimization procedure that finds the region R within the set of all rectangular subregions in the image maximizing a given quality function f. The **branching** step splits the side of a set of boxes into halves, and the **bounding** step checks if the quality bound $\hat{f}$ of f is within a preset threshold so that it is still for an optimum to exist in the current set of boxes. If not, all the boxes belonging to this particular branch can be safely discarded, and the savings significantly speeds up the search process. We perform this recursive procedure between the query ROI and all boxes in all images contained in our dermoscopic database. Out of the various bounding functions proposed in [61], We found the histogram intersection kernel works well

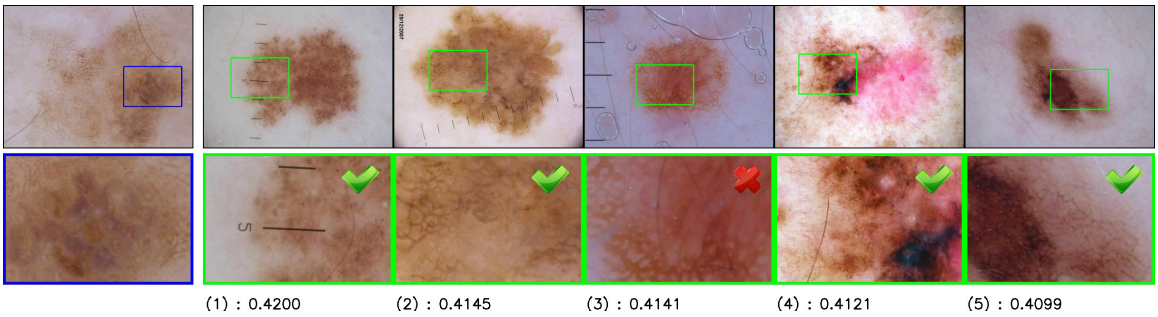
for our purpose. Hence, the bounding function f is $HI(h^Q, h^R) = \sum_k \min(h_k^Q, h_k^R)$ where h^Q and h^R denote the *bovw*-histogram of the query region Q and subimage R, respectively. The retrieval task now becomes to identify the N image regions with highest similarity to Q. In general, ESS is able to find the global optimum with $\mathcal{O}(n^2)$ runtime for a $n \times n$ image, and it performs sub-linearly over a database of such images. This enables DermFind to perform ROI search interactively.

Qualitatively, we can also see the effectiveness of the relevant case retrieval in Fig. 48 and 49. In a typical image search (Fig 48(c)), An query image is given in the first column. DermFind identifies the closest neighbors of the image based on the global (entire image) statistics as discussed in Section 7.1.3. However, in some scenario, such as a lesion with compound appearance as shown here, the user may want to be able to search for relevancy based on a part of the image. DermFind provides such flexibility to the user through interactive ROI search. In both Fig. 48(b) and Fig. 48(c), respectively, the user draws a box (blue) around the area of interest and DermFind is able to find relevant cases that match this partial appearance encompassed by the green boxes. The particular appearance in ROI in Fig. 48(b) corresponds to a dermoscopic features called *broadened network* and DermFind returns a number of cases with the same feature (The green check-mark on the top right corner of the enlarged ROI view indicates that these are indeed broadened network according to annotation ground-truth). The feature shown in Fig. 48(c), fibrillar pattern, is rare, DermFind is able to find similar regions by leveraging a large database of images though making some mistakes along the way. Figure 48(d) and (e) shows the results from similar ROI searches on discrete network and follicular openings, respectively.

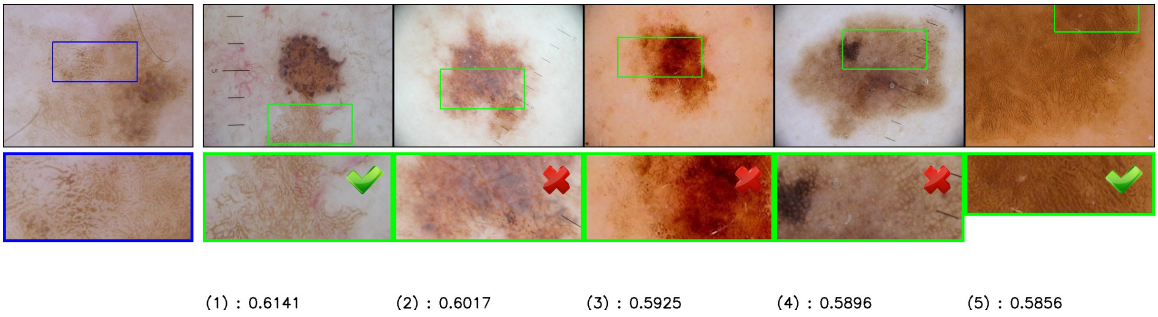
Figure 49(a) - (c) show ROI search results for some of the most commonly observed dermoscopic features: (a) parallel furrow patterns, (b) globular pattern, and (c) telangiectasia (pinpoint vessels). Notice that in Fig. 49(b) and (c), both globular



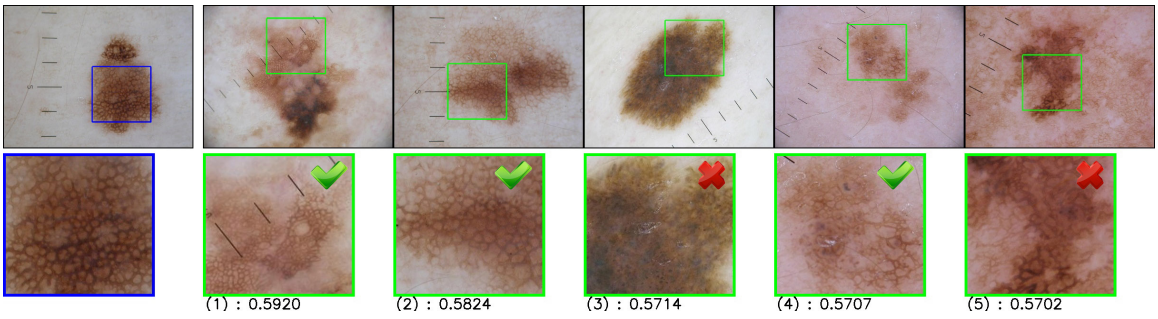
(a) Typical image search



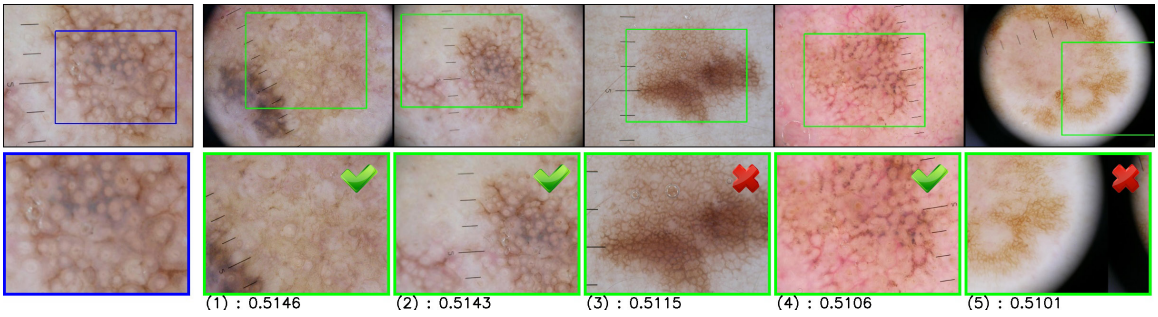
(b) ROI search: blue white veil



(c) ROI: fibrillar pattern



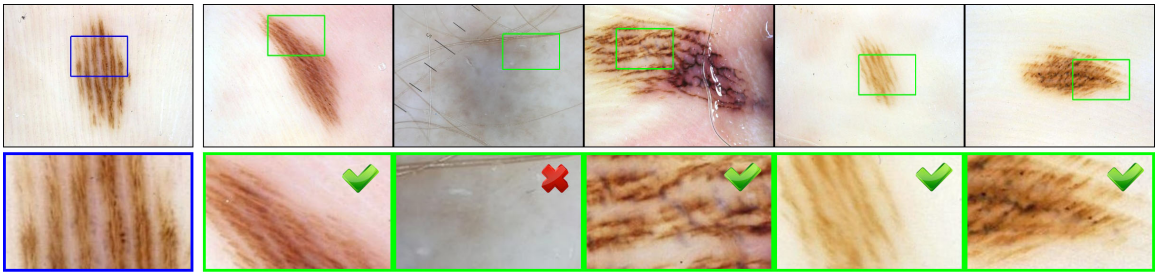
(d) ROI: discrete network



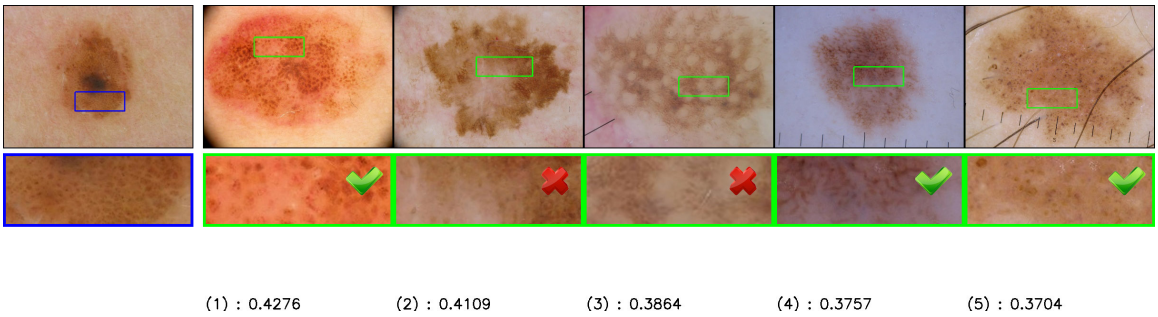
(e) ROI: follicular openings

Figure 48: Example of ROI search.

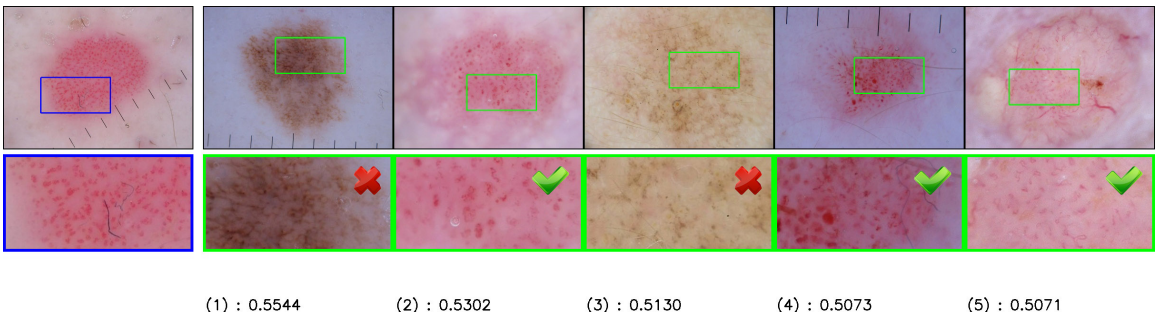
pattern and pinpoint telangiectasia share similar appearances with light discrete network and follicular opening (The incorrect returns marked by a red cross on the top right corner), and the task of distinguishing them still remains challenging. Nevertheless, our results indicate that, using a unified *bovw* framework and region-of-interest search capability, DermFind is able to bring some success in this aspect. Quantitatively, we achieved a 73.6 % accuracy for a test trial of 40 ROI searches returning 5 closest neighbors.



(a) ROI: parallel furrow pattern



(b) ROI: globular pattern



(c) ROI: telangiectasia (pinpoint vessels)

Figure 49: Example of ROI search.

7.3 Relevance feedback for exemplar-based CBIR-CAD

In a typical CBIR systems [21, 22, 12], a search consists of a query followed by repeated instances of user feedback, where the user provides some form of comments or actions based on the exemplars that are retrieved. The system can then incorporate such feedback to modify the query or search criteria in order to refine the retrieval results. Exemplar-based analysis is particularly well-suited to query refinement due to its interpretability and the ease with which it can be incorporated into an interactive system. These properties make exemplar-based analysis an especially attractive choice for CBIR-CAD systems. Exemplar-based analysis can be designed to incorporate user feedback through direct manipulation of retrieved exemplars. In contrast, to provide such feedback in the case of model-based analysis often requires the user to have additional system-specific knowledge in order to adjust specific system parameters. In addition, the ability to examine the exemplars may, over time, enable an experienced user to gain greater understanding of the cause for a particular recommendation. Such insight is much harder to obtain for black-box methods.

DermFind has a set of basic feedback mechanisms that are commonly seen in conventional CBIR systems. These include a standard ranked *matrix of images* display (the right window in Fig. 50), which displays retrieval results ranked by their relevance to the query, and an interface for feature specific user feedback control; in a separate window, retrieval results are displayed according to their rankings with respect to individual features (the middle window in Fig. 50). By adjusting the contribution of each feature, the user can interactively modify the search criteria and update the retrieval results. However, the effect that adjusting individual feature contributions may have on the retrieval refinement can often be difficult to interpret, and the interactive search for the best criteria becomes a trial-and-error type of guesswork. As a result, I discover that many dermatologists simply shy away from dealing with these controls after a few initial trials that return unsatisfactory results. For

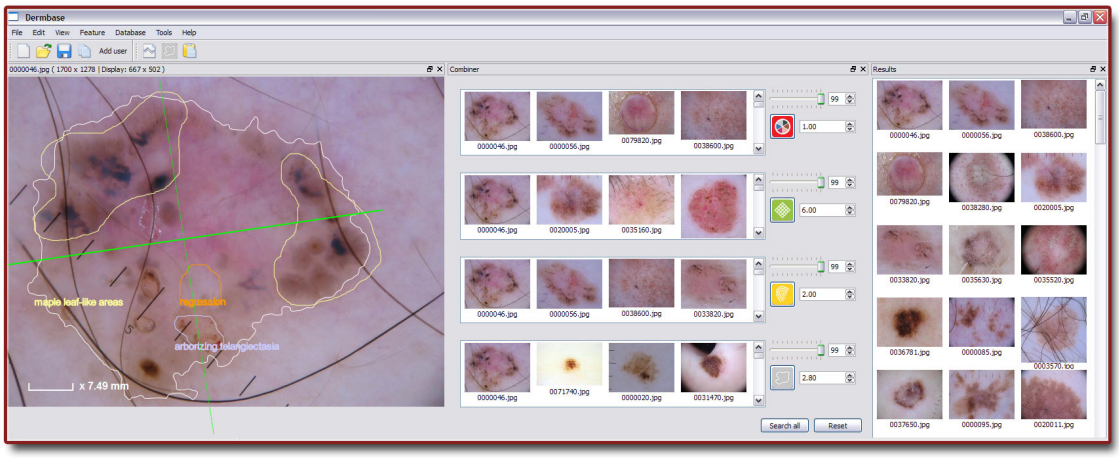


Figure 50: Basic feedback mechanism for search with compound criteria.

an expert-support CBIR system like DermFind, the ability to incorporate experts' feedback to refine retrieval results is of most importance. Therefore, I resort to the *relevance feedback* (RF) mechanism, which provides an efficient framework for mapping high-level semantics or domain specific knowledge to low-level feature representation by exploiting users' interaction.

7.3.1 Relevance feedback in DermFind

Relevance feedback was originally developed in text-based information retrieval (*IR*) systems [86]. Early in the IR community, researchers realized that users might have difficulty converting their need for information into a request, but they could recognize useful information when they saw it. This lead to the notion of relevance feedback: users mark exemplars returned by an initial search as relevant or non-relevant to their needs and present this information to the IR system. The system uses this information to produce a modified version of the query, which is then used to improve the retrieval performance, both quantitatively - retrieving more exemplars like the relevant ones - and qualitatively - retrieving exemplars similar to the relevant ones before others. This process is known as an iteration of relevance feedback. Through such refinement iterations, relevance feedback provides an efficient framework for mapping high-level

semantics to low-level features by exploiting user's interaction in retrieval systems.

There are four main models of retrieval in a typical IR system: *Boolean*, *vector-space*, *probabilistic*, and *logical*. Relevance feedback mechanisms exist for each of these four models.[3] In the context of DermFind, I have found that the vector-space model is particularly well-suited to the *bovw* dermoscopic feature representation.

In the vector-space model, a document[4] is represented by a vector of n weights: $\vec{e} = (e_1, e_2, \cdots, e_n)$, where n is the number of unique terms in the document. Queries are also represented in this format, and the similarity (e.g. inner product, cosine correlation, histogram intersection, etc.) of the document vectors to a query vector gives a retrieval score to each document.

In the context of DermFind, a dermoscopic image is a document and a dermoscopic feature is a term. Thus, relevance feedback mechanisms that have been developed for the vector-space model in the IR community can be naturally incorporated into the retrieval processes described in Section 7.2. Two major strategies exist for implementing relevance feedback for the vector-space retrieval model: query point movement strategy, which re-defines the query by moving it in the direction of relevance inside the feature space, and similarity measure updating strategy, which updates feature weights to reflect their different degrees of importance. The similarity updating strategy typically incurs more computational cost and may hinder the interactivity requirement of the DermFind system; therefore, I focus exclusively on the query point movement strategy.

7.3.1.1 Rocchio's fomula

Rocchio's relevance feedback formula (Eq.11), which was introduced in [82], was the first query point movement scheme developed on top of the vector-space model. It

[3] We refer interested readers to [86] for a more detailed description of these models.

[4] We adhere to terminologies used by the IR community and use the term *document* loosely to mean an entry of the database

defines the problem of retrieval as that of generating an optimal query; one that maximizes the difference between the average vector of the relevant documents and the average vector of the non-relevant documents:

$$\vec{q_m} = \vec{q_0} + \frac{1}{|E_r|} \sum_{\vec{e_i} \in E_r} \vec{e_i} - \frac{1}{|E_{nr}|} \sum_{\vec{e_j} \in E_{nr}} \vec{e_j}, \tag{11}$$

where q_0 is the initial query vector, q_m is the new query vector, and E_r, E_nr are collections of relevant and non-relevant exemplars, respectively. During one RF iteration, the query vector q_0 is updated, moving it closer to the mean of the set of relevant exemplars E_r, and further from the mean of the non-relevant exemplars E_{nr}. This is accomplished by the addition and re-weighting of query feature terms to reflect their utility in discriminating relevant from non-relevant exemplars.

In DermFind, I adopt Rocchio's modified vector-space RF formula introduced in [49] because it weights the relative contribution of the original query, relevant and non-relevant exemplars, which gives the user one more level of control. In equation 12, the α, β, and γ values specify the degree of effect of each component on relevance feedback.

$$\vec{q_m} = \alpha\vec{q_0} + \beta\frac{1}{|E_r|} \sum_{\vec{e_i} \in E_r} \vec{e_i} - \gamma\frac{1}{|E_{nr}|} \sum_{\vec{e_j} \in E_{nr}} \vec{e_j} \tag{12}$$

7.3.1.2 Interface and user interaction

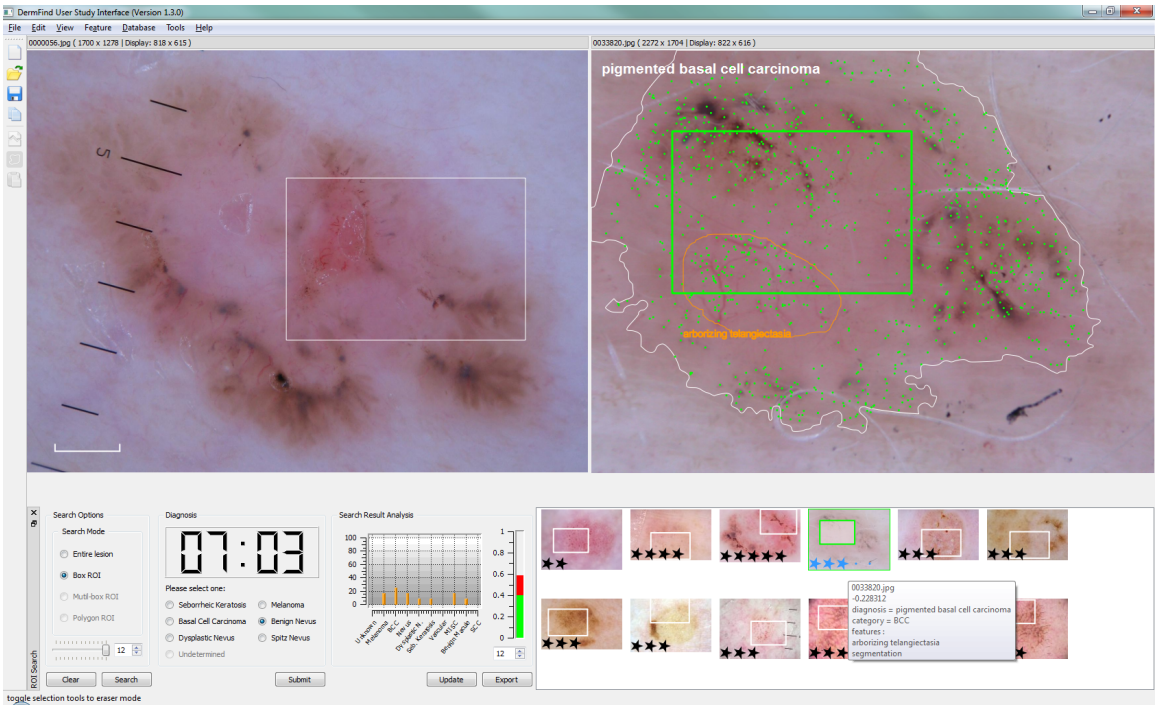
To facilitate Rocchio's relevance feedback mechanism, I have designed an intuitive interface in the context of DermFind's standard ranked *Matrix Of Image* display (bottom right window shown in Fig.51(a)). After a query (top left image in Fig.51(a)) is submitted for a search, DermFind returns an initial set of exemplars, each with a default rating of three stars as shown in Fig.51(b). There are five levels of relevance presented in this star rating system[5], with one star marking "not relevant at all" and

[5]We employ the five level star rating system because it is the dominant user rating feedback paradigm, and users are likely to have seen it.

five star marking "very relevant". The user can adjust the star rating of any exemplar based on her subjective belief of how relevant the exemplar is to the query. The star rating is then translated into coefficients of the Rocchio's modified formula: $\gamma = 0.25$ for one star (-2), $\gamma = 0.125$ for two stars (-1), $\beta = 0.25$ for four star (+1), and $\beta = 0.5$ for five stars (+2). The three-star rating marks the exemplar as neutral and excludes it from the feedback update.

After the user completes her rating, DermFind computes a new query vector q_m based on the initial query q_0, the relevant set E_r and the non-relevant set E_nr using Rocchio's modified formula (Eq.12). An search using q_m will return a different set of exemplars, and result in an update of the diagnosis based on statistics of the returned exemplars.

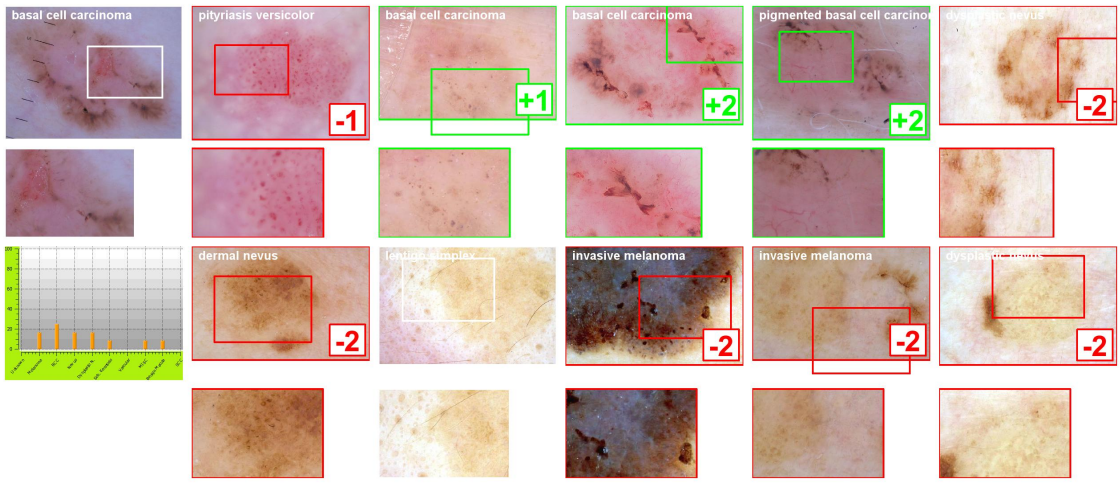
Figure 52 illustrates two iterations of the relevance feedback refinement. In Fig.52(a), the query image resides in the top left corner with a zoomed-in region-of-interest displayed below it. The ten retrieved exemplars are listed on the right, ordered from left to right, top to bottom, by their similarity to the query. The analysis chart (lower left corner) indicates that the returned exemplars carry various diagnoses. The user has already examined the details of returned region-of-interests and assigned a relevance rating, shown in the lower right corner of each exemplar, with green indicating relevant, red non-relevant, and white neutral. These ratings are then incorporated into the RF update described above, and an updated list of exemplars are shown in Fig.52(b). This time, the diagnoses these exemplars carry start to converge to either BCC (Basal Cell Carcinoma) or dysplastic nevus. Another round of user examination results in a set of new relevance ratings, which gets incorporated into the next round of relevance feedback update (Fig.52(c)). This time, the diagnoses finally converge to basal cell carcinoma, which is correct according to the ground-truth diagnosis of the query.



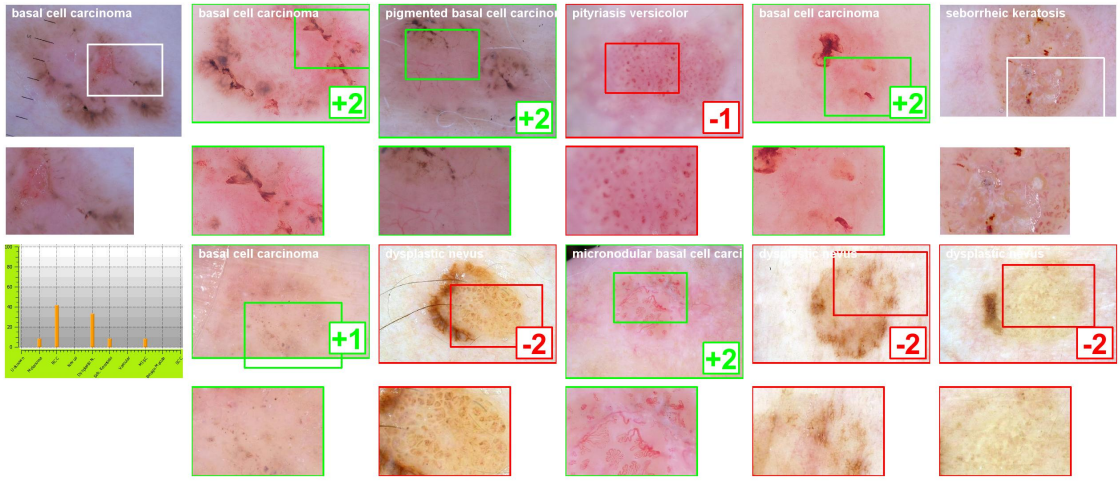
(a)

(b)

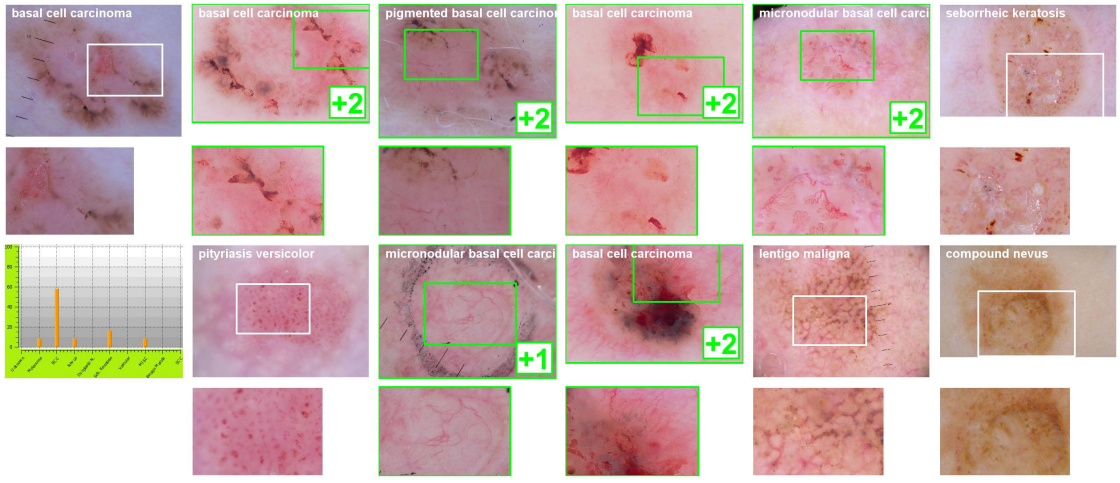
Figure 51: (a) DermFind relevance feedback interface, (b) five level relevance feedback star ratings.



(a)



(b)



(c)

Figure 52: Illustration of relevance feedback in action (a) intial search results and relevance star ratings assigned by the user, (b) refined search results after the first round of relevance feedback update, (c) further refined results after another round of relevance feedback update.

The above is a demonstration of how user's interaction with the DermFind relevance feedback interface can lead improved search results. The utility of the proposed relevance feedback interface is closely tied to the performance of the overall system. I will describe the evaluation of this interface in the context of DermFind in the user study section 8.2.

CHAPTER VIII

EVALUATION OF THE DERMFIND SYSTEM

We evaluate the effectiveness of the DermFind system both quantitatively and qualitatively. Quantitatively, we look at DermFind's performance as a computerized diagnosis system . Qualitatively, we measure the effectiveness of the DermFind system to clinicians in skin lesion diagnosis decision support through the user study. In the ensuing sections, We will describe these evaluations in detail respectively.

8.1 Pigmented skin lesion diagnosis

The DermFind system currently stores a total of 2533 dermatological cases, among which 1570 have confirmed pathological diagnoses. The distribution of the diagnoses of these cases are shown in Fig 3.3.1(b). For quantitative evaluation, we consider the following two scenarios of search assisted diagnosis.

8.1.1 ABCD analysis

There are several standard methods in dermatology for assessing abnormality in pigmented skin lesions such as the ABCD rule, 7-point checklist, and Menzie's method. These methods provide a standardized checklist for assessing abnormality in color, shape (symmetry), texture, and structural characteristics of the lesion [1]. DermFind adapts the ABCD rule as an automatic diagnosis mechanism because it is a standard pattern analysis scoring system used by clinicians to identify melanocytic lesions which are at risk for melanoma. A detailed discussion of the ABCD rule is in Section 7.1.2. In summary, the ABCD rule uses four criteria: **A**symmetry, **B**order abruptness, and **C**olor, and **D**ifferential structures to generate a Total Dermoscopy Score (TDS), which is computed as $TDS = A \times 1.3 + B \times 0.1 + C \times 0.5 + D \times 0.5$. A

lesion with $TDS > 5.45$ is marked as melanoma according to the rule. We computed the TDS on DE1 (Table 3.3.1): a set of 1269 melanocytic lesions (405 melanoma and 864 nevi) in our database, using methods described in 8.1.1 to determine the A, B, C scores, and using the dermoscopic feature descriptor to calculate the D score. We then mark any lesion with with $TDS > 5.45$ as melanoma according to the ABCD rule. Comparing to the ground truth pathological diagnosis, our computer-aided ABCD analysis achieves an accuracy of 80.3% with a 21.1% false positive rate. Observing these results closely, we find that the false positives are mostly nevi which are categorized as dysplastic, which is arguably considered as a transitional state between completely benign nevi and melanoma. The fact that our ABCD analysis makes most mistakes on these cases is expected as dysplastic nevi are often sources of confusion for physicians too in their practices.

8.1.2 Malignancy detection via retrieval

While standard methods such as the ABCD rule is popular amongst certain group of dermatologists, they are far from being adopted universally, many experienced physicians often check a number of color-texture cues simultaneously during the examination of a skin lesion. To simulate this, our evaluation considers a particular scenario of search assisted diagnosis. Given a query image, we want to determine if the lesion in it is benign or malignant. We base our decision on images that look similar to the query. To evaluate the effectiveness of such an approach, we perform a compound search (See Section 7.1.3 for a detailed discussion) for each image in the database and measure the cumulative accuracy. In this evaluation, the search is defined by three equally weighted criteria: layered color descriptor, sPACT, and shape represented by the Hu moment vector. By varying the number of neighbors we use as informants, a tentative diagnosis is suggested by the system using the majority rule. This diagnosis is compared with the confirmed pathological diagnosis to produce the

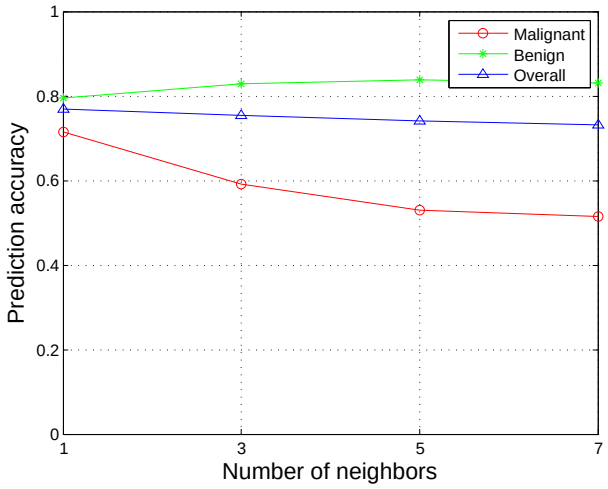


Figure 53: Malignancy detection via retreival accuracy graph.

accuracy curve (Fig. 53). The prediction accuracy of malignancy for a query image from its retrieved nearest neighbors is 76.94%.

8.1.3 Diagnosis via retrieval

We measure the retrieval performance on DE2 (Table 3.3.1): the set of all dermoscopy images with confirmed pathological diagnoses using the standard precision-recall (PR) curve. There are 1617 such images from 1570 dermatological cases since about some cases are associated with multiple images.[1]

There are nine categories of diagnoses we use for this evaluation. the distribution of diagnoses of all 1617 images is listed in Table 5. The feature I use is the *bovw* representation with DIP descriptors in the RGB color space. I have tried a number of cluster sizes and found 2000 to give the best performance, so each feature vector is of length 2000. A similarity score is calculated between the query and each retrieved exemplar using the normalized histogram intersection measure

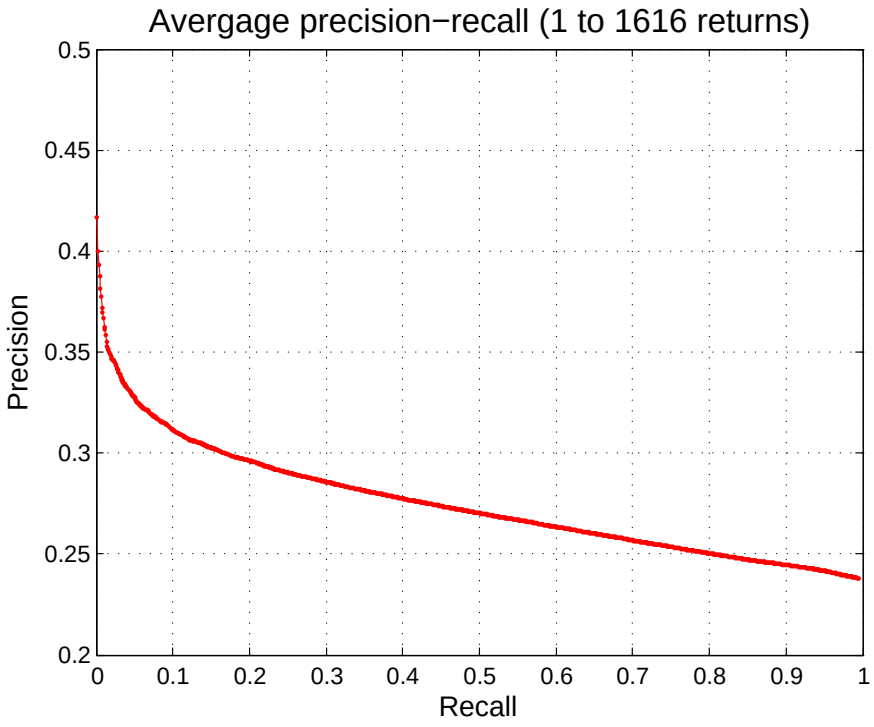
[1] A case can be associated with multiple images in following situations: 1) picture of the same lesion were taken during mutliple visits, usually for re-evaluation of the lesion. 2) A lesion is too large to fit into the scope view, so several pictures were taken to cover the entire lesion. 3) Multiple pictures were taken to zoom in on some minute details in a large lesion.

Table 5: The distribution of pathological diagnoses of all the images in the DermFind database.

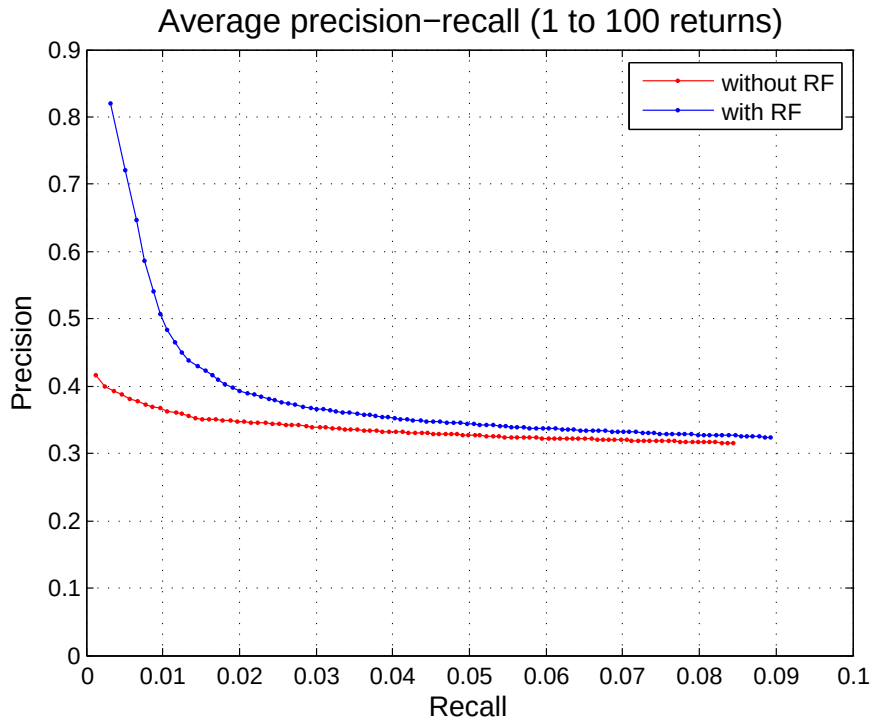
Number of	Number
melanoma	416
basal cell carcinoma	88
nevi	269
dysplastic nevi	595
Seborrheic keratoses	96
vascular lesion	60
misc	46
benign macule	36
squamous cell carcinoma	11

$NHI(h^Q, h^R) = \sum_k min(\frac{h_k^Q}{\|h^Q\|_1}, \frac{h_k^R}{\|h^R\|_1})$; this score is then used to rank the return exemplars, which are considered relevant if they share the same diagnosis as the query. I measure precision-recall for all 1617 images as queries and plot the average PR curve in Figure 54(a). Precision is around 0.24 when recall is near 1; this is what the retrieval precision would be if I just randomly rank the exemplars. The precision of the first several of returns is significantly higher (over 0.4) indicating that DermFind is indeed capable of retrieving relevant exemplars. If we were to use the diagnosis associated with the first returned example to label the query, we would get an average diagnosis accuracy of 42%. To put this in perspective, our user-study reveals that the average accuracy of giving specific diagnoses (in contrast to detecting malignancy) to lesions in dermoscopy images is around 50% even for experienced dermatologists.

The precision-recall measurement was also used to quantitatively measure the benefits of introducing relevance feedback into the retrieval process. However, quantitative evaluation of such in a realistic setup - asking users to provide relevance feedback for a large set of sample images, would require a significant amount of user involvement, which was not possible within the time frame of this study. As an alternative, we instead simulate a relevance feedback scenario for every image in the dataset. Given a query image, the top six returns of the initial retrieval were automatically labeled by the system as either relevant or irrelevant based on the ground-truth



(a)



(b)

Figure 54: Average precision-recall curve a) for retrieving exemplars with same diagnoses using all 1617 images as queries, b) precision-recall for retrieval with relevance feedback (blue) and without relevance feedback for the first 100 returns.

diagnoses.[2] The initial query vector was then modified using Rocchio's modified formula (Eq. 12) with $\alpha = 1$, $\beta = 0.25$, and $\gamma = 0.125$, to incorporate the relevance information. The new query vector became the input of a second round of retrieval, the results of which were used for measuring the performance of retrieval with simulated relevance feedback. We have plotted the average precision-recall curve (blue in Figure 54(b)) by varying the number of returned exemplars. Comparing this to the retrieval performance achieved without using relevance feedback (red curve), we can see a drastic improvement in the top ranked returns, which remains significant even after the first six returns. This indicates that the presence of a relevance feedback mechanism indeed improves the retrieval performance, both quantitatively - retrieving more exemplars like the relevant ones - and qualitatively - retrieving exemplars similar to the relevant ones before others.

[2]The simulation assumes that users always modify the first six returns, and beyond that, they lose their patience and move to the next query.

8.2 DermFind user study

We designed an experiment to quantify the overall benefit of Interactive Search-Assisted Decision Support from DermFind for skin lesion diagnosis. To achieve this, we compared the diagnosis performance achieved by qualified participants using the DermFind interface in two modes. Mode 1 is the **S**earch mode, which gives users access to the DermFind system with fully-functional interactive search capabilities (including ROI retrieval and relevance feedback). The users assigned to perform diagnosis using this mode are considered to be the experimental group. Mode 2 is the **B**rowsing mode, which disables all the search functionalities. The interface acts as a basic dermatological case image browsing system. This mode simulates the process of flipping through case archives which is the common strategy when dermatologists encounter difficult cases. The users are considered to be the control group. By comparing participants' diagnosis performance under both modes, we quantify the benefit of the interactive search component of the DermFind system in providing diagnosis support.

8.2.1 User interface

The two modes for the user study are distinguished by the presence and absence of the interactive search capability. In all other regards, the user interface is the same between the two modes in order to avoid possible user experience bias. The common layout (shown in Fig. 55) of both modes includes a query image view that occupies the top left side of the interface and a comparison image view on the top right side. The bottom part of the interface consists of (from left to right) 1) search option panel, 2) timer and diagnosis submission panel, 3) result analysis panel, and 4) retrieved image panel.

In search mode (Fig. 56), the search option panel allows user to select different search options (entire lesion, box ROI, or multi-box ROI), adjust the number of

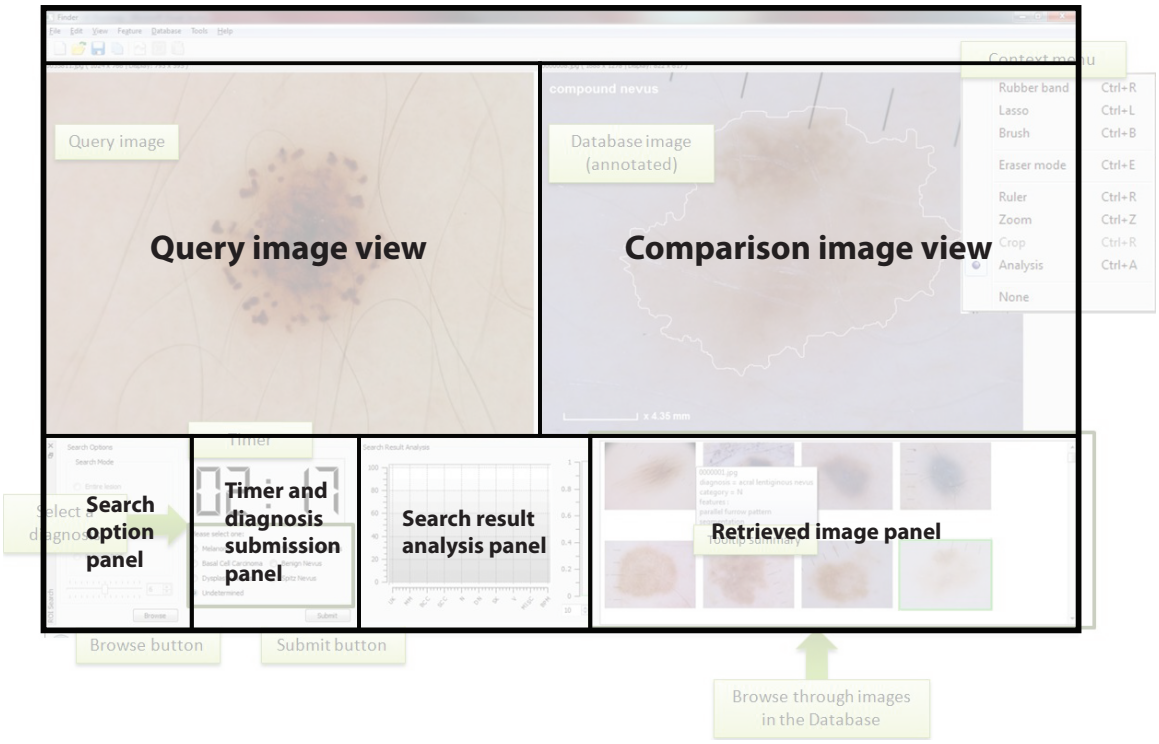
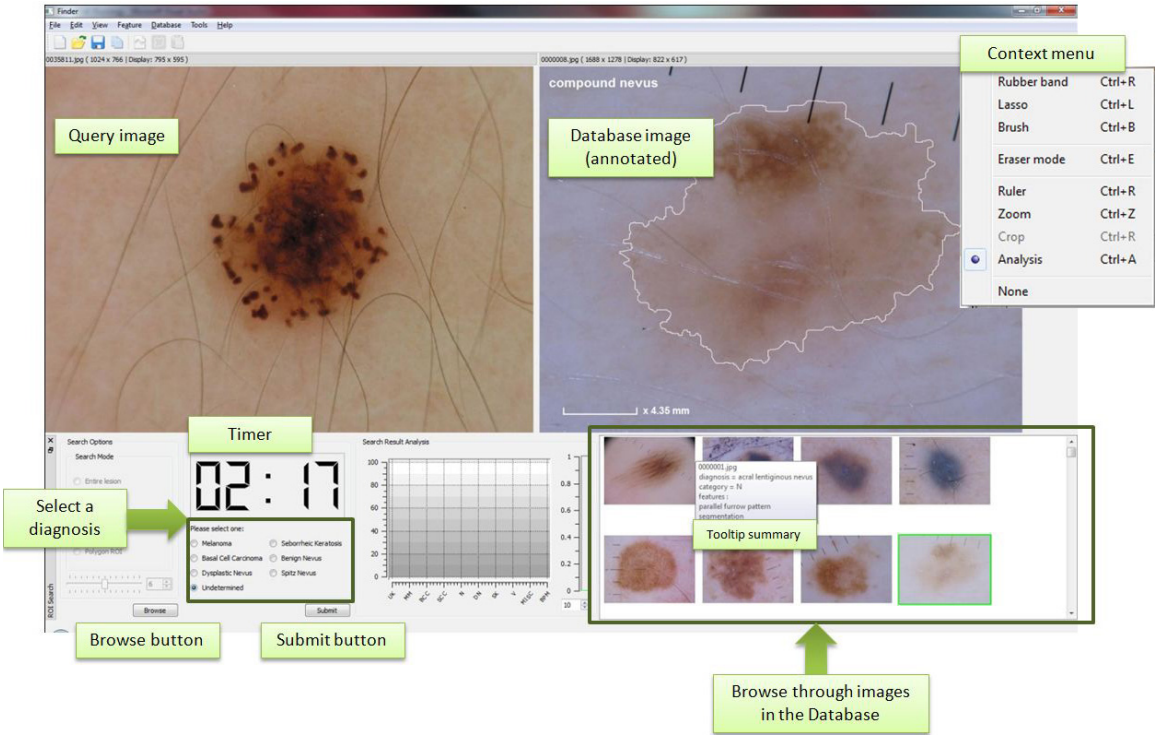
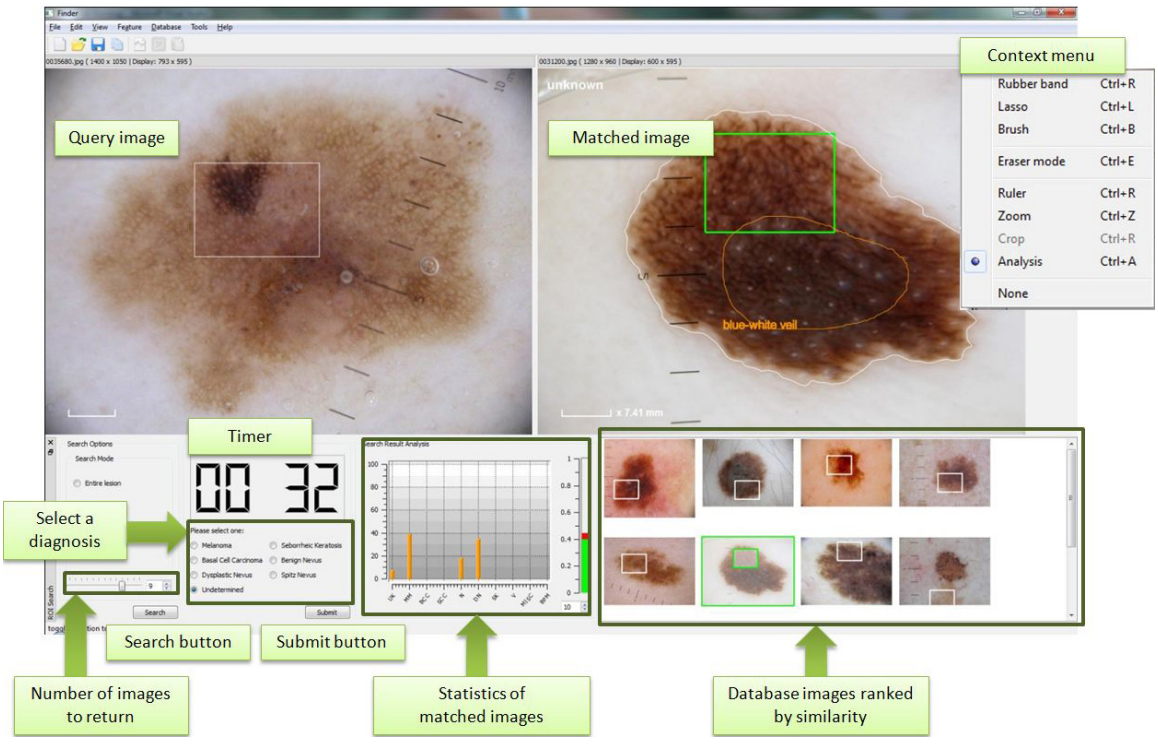


Figure 55: User study interface.

exemplars N to be returned using the slider, and initiate a search. The panel to the right of the search option panel shows a count-down timer that reminds the user how much time is left before he/she has to submit a diagnosis. There are a total of six categories of pigmented skin lesions included in the user study: melanoma, seborrheic keratosis, basal cell carcinoma, benign nevus, dysplastic nevus, and Spitz nevus. The "benign" radio button is automatically checked if the participant does not select a diagnosis within the allocated time. The retrieved image panel at the bottom-right of the display shows the N exemplars returned from the database based on the chosen search option. The result analysis panel to the left of this display shows diagnosis statistics collected from these returned exemplars. Hovering on an exemplar image brings up a tool-tip summary of the case annotation, which includes a detailed diagnosis, diagnostic category, and a list of the dermoscopic features that are present. Left-clicking on the image focuses on the exemplar and brings it up to the comparison image view, enabling the participant to perform a detailed examination.



(a)



(b)

Figure 56: User study interface : (a) baseline browsing mode, (b) interactive search mode

Table 6: User study scheme

Images	Group 1	Group 2
1-12	Browsing mode	Search mode
13-24	Search mode	Browsing mode

In browsing mode the interface is the same, except that the search option panel and the search result analysis panel are both disabled. The search button is also replaced with a browsing button, which, upon being pressed, brings up all images [3] within our database with corresponding annotation.

8.2.2 User study design

The study required the participants to have sufficient experience to make diagnoses of pigmented skin lesions from dermatological images. Therefore, we recruit practicing dermatologists, dermatological residents, experienced dermatology physician's assistants, or medical students who have studied dermatology. Given the inherently-limited number of sufficiently-qualified participants, I devised an alternating scheme in which each participant performed the experiments in both the control and the experimental groups. Each participant was randomly assigned to one of the two groups (Table 6). Group 1 users performed diagnosis of the first twelve images under browsing mode, and then switched to search mode to diagnose the remaining twelve images. Group 2 users did this in the reverse order. In this manner, the diagnoses from the participants were collected in both modes, making sure that all cases are diagnosed an equal number of times in each mode.

I have recruited ten participants from Emory University and the Atlanta VA Medical Center for this user study. The studies took place at various locations including the Computer Perception Lab on the second floor of the Technology Square Research Building on Georgia Tech Campus, office space at the Atlanta VA Medical Center,

[3] except for the query images

and the lounge area at the Emory Woodruff Memorial Research Building. All participants performed their diagnosis using the DermFind interface on a computer running Windows 7 operating system.

8.2.3 Procedures

This section describes the procedures that were used in the DermFind user study. At the beginning of the study, each participant received a written instruction of the tasks he/she had to perform and a verbal explanation of the procedures. The participant then read and signed the consent form (Appendix C) and filled out a short background survey (Appendix D) regarding his/her credentials and experience in diagnosing pigmented skin lesions.

The following is a list of steps that each participant follows: Each participant was required to

1. listen to the instructions on how to use both the basic browsing mode and DermFind's interactive search mode,

2. take as much time as needed to practice in the training session using both modes,

3. follow the procedure laid out in Table 6 and complete the user study (will be elaborated below).

4. complete a questionnaire (Appendix E) regarding the experience of using the DermFind system after finishing the experiment.

5. listen to a debriefing session containing details on the purpose of the experiment and how their data will be used.

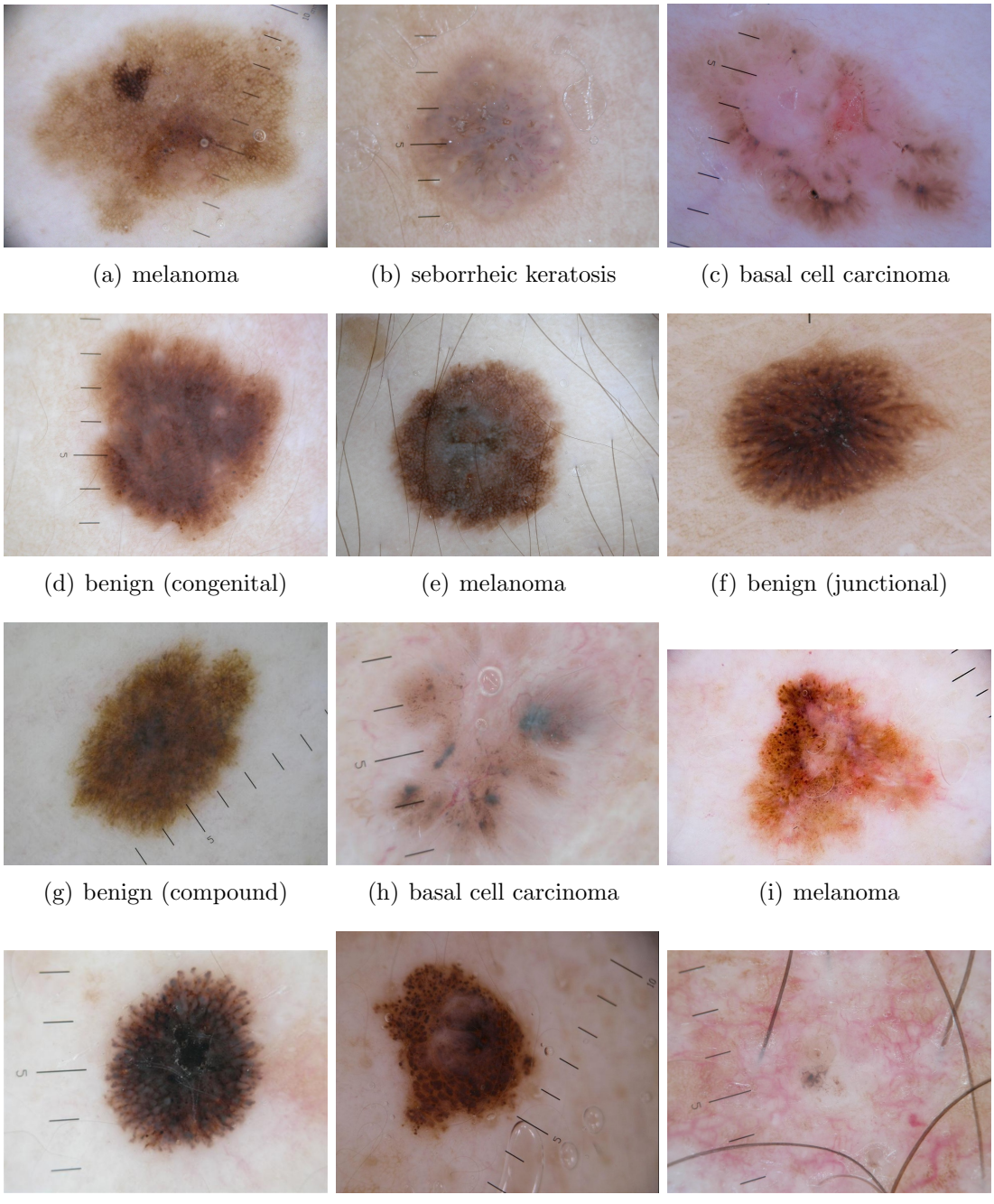
Here we elaborate on point 4). During the experiment, a set of 24 dermoscopy images (Dataset DU1 in Table 3.3.1), which are selected from the DermFind case repository with confirmed ground-truth pathological diagnoses, are displayed in DermFind's

user study interface (illustrated in Fig. 55 and 56). These images were presented to each participant in a pre-determined (randomly) order in two and half-minute intervals. For each image, the participant was asked to select one diagnosis from a list of six possible diagnoses or mark the case as undetermined. If the allocated time had passed before he/she could make a decision, the case was reported as benign (the default diagnosis) and the system moved on to the next case. The experiment ended when all 24 case images were diagnosed. The actual diagnosis of each case and the time it took the participant to reach a conclusion were recorded by the system.

8.2.4 Evaluation

8.2.4.1 Grading of participants' performance

Although the pathological diagnosis for each of the dermatological images used in the user study was definite, ambiguity arose from the fact that the participants were only given access to the lesions' appearance. The inability to examine the lesions' physical properties, such as raisedness, and consult the patients' information and medical history made it very difficult for even veteran dermatologists to give a diagnosis for some of the lesions. In addition, there are still disagreements over the classification of certain skin lesions even inside the medical community. For example, some dermatologists treat dysplastic nevus as an early stage of melanoma, instead of a separate diagnosis. Subtle differences between the two lesions illustrated in Fig. 58(h) and Fig. 58(i) carry diagnoses of melanoma and benign nevus, respectively. However, based on their appearance alone, they both are quite often categorized by many dermatologists as dysplastic nevus. This motivated me to grade participants' diagnosis accordingly. For example, in the case of Fig. 58(h), a diagnosis of melanoma receives full credit: a score of 100, and dysplastic nevus receives half credit. Similarly for Fig. 58(i), an answer of benign nevus warrants full credit while dysplastic nevus gets half. Using this grading scheme, we obtained the final performance score sheet as illustrated in Figure 59.



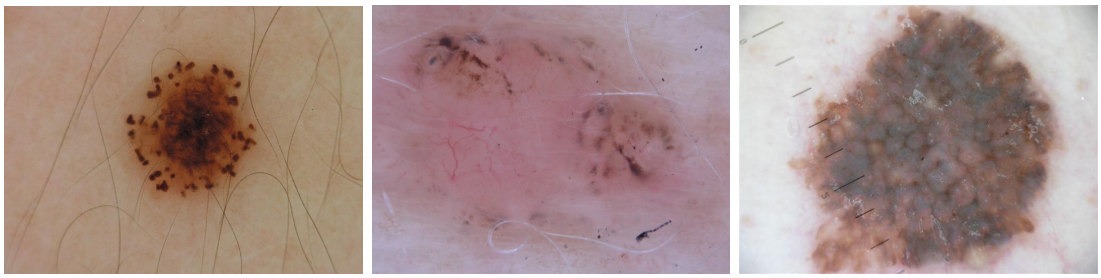
(a) melanoma (b) seborrheic keratosis (c) basal cell carcinoma

(d) benign (congenital) (e) melanoma (f) benign (junctional)

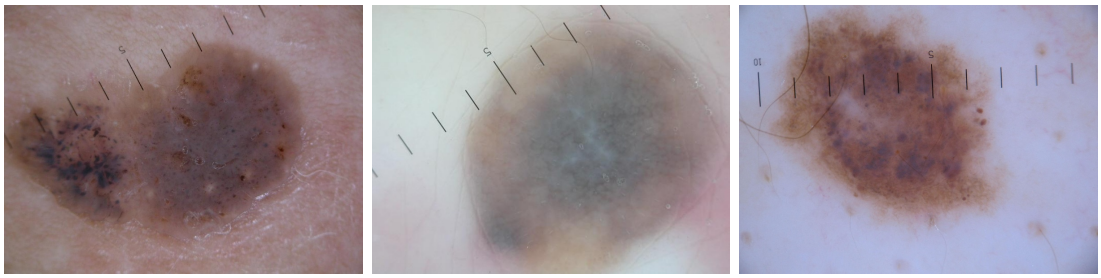
(g) benign (compound) (h) basal cell carcinoma (i) melanoma

(j) Spitz nevus (k) dysplastic nevus (l) basal cell carcinoma

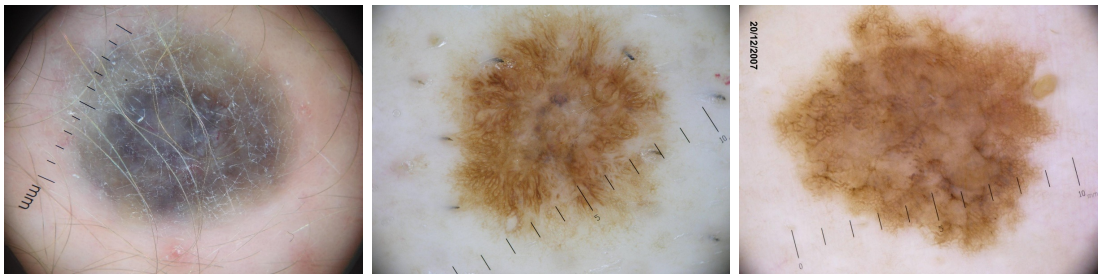
Figure 57: Example case images used in the DermFind user study: 1 - 12



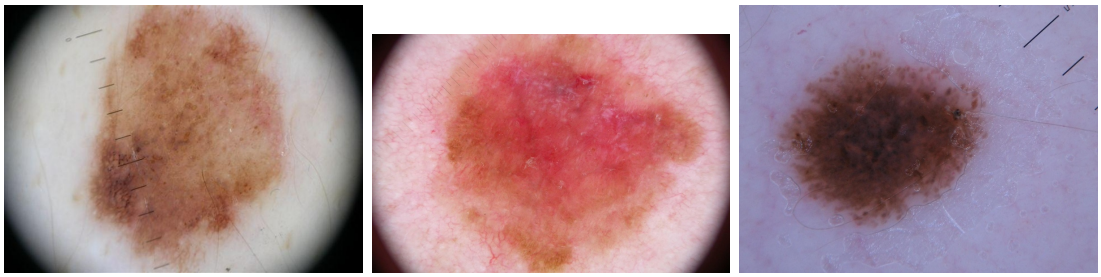
(a) dysplastic nevus (b) basal cell carcinoma (c) seborrheic keratosis



(d) seborrheic keratosis (e) benign (blue) (f) dysplastic nevus



(g) benign (blue) (h) melanoma (i) benign (compound)

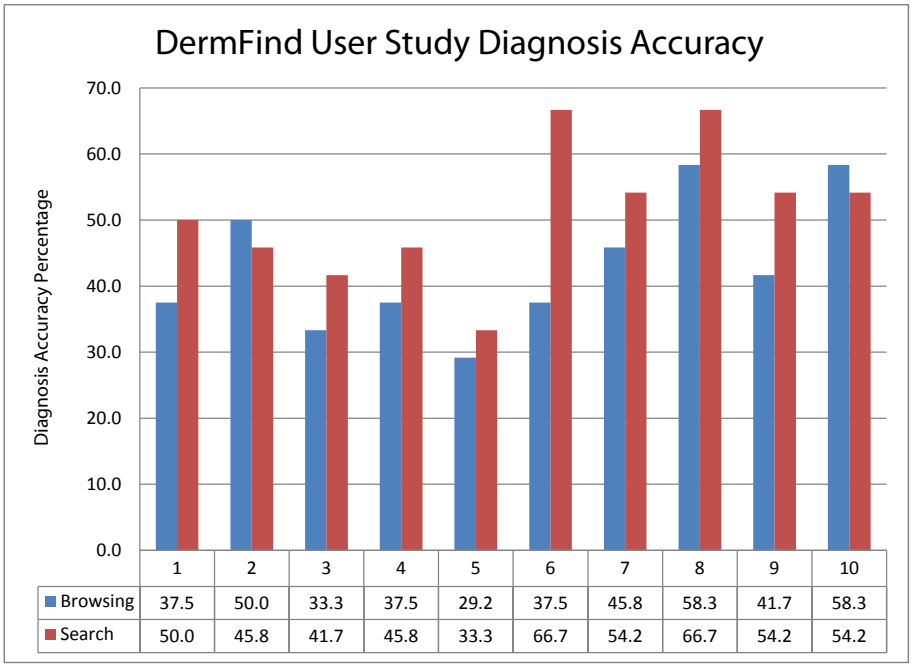


(j) dysplastic nevus (k) melanoma (l) benign (junctional)

Figure 58: Example case images used in the DermFind user study: 13 - 24

Index	Users										Case
	1	2	3	4	5	6	7	8	9	10	Average
Group:	2	2	1	1	2	1	2	1	1	2	
0	50	100	100	50	100	0	100	100	0	50	65.0
1	0	50	50	50	50	0	50	50	50	100	45.0
2	100	0	0	0	100	0	0	0	0	100	30.0
3	50	0	50	50	0	100	50	50	100	50	50.0
4	0	100	100	0	0	0	100	0	100	50	45.0
5	100	50	0	0	0	50	50	100	0	50	40.0
6	100	50	0	100	50	50	50	100	50	50	60.0
7	0	0	0	0	0	100	0	100	0	0	20.0
8	50	50	50	0	100	0	100	50	100	100	60.0
9	50	100	50	50	0	50	50	50	50	0	45.0
10	100	50	0	50	0	0	100	0	50	0	35.0
11	0	0	0	100	0	100	0	100	0	100	40.0
Average:	50.0	45.8	33.3	37.5	33.3	37.5	54.2	58.3	41.7	54.2	44.6
12	0	100	0	0	0	100	100	100	100	50	55.0
13	0	0	0	0	0	100	0	100	100	100	40.0
14	50	50	50	100	50	100	0	100	100	0	60.0
15	0	50	50	100	100	100	0	0	0	0	40.0
16	100	50	100	50	0	50	100	100	50	100	70.0
17	50	100	50	0	100	50	0	0	50	100	50.0
18	0	100	50	50	0	100	50	50	0	100	50.0
19	100	50	50	100	50	100	50	50	50	50	65.0
20	50	50	50	50	0	50	50	100	50	50	50.0
21	100	50	0	0	0	0	100	100	100	100	55.0
22	0	0	0	0	0	0	0	50	0	0	5.0
23	0	0	100	100	50	50	100	50	50	50	55.0
Average:	37.5	50.0	41.7	45.8	29.2	66.7	45.8	66.7	54.2	58.3	49.6
Search:	51.3										
Browsing:	42.9										

Figure 59: User study data sheet: The second row shows the subjects' ID. The third row shows their group IDs (See Table 6), followed by twelve grades for the first set of images, then the average, grades for the second set of twelve images after mode switching, and then the average of that. The last column of the table displays the average grades from the participants for each image.



	1	2	3	4	5	6	7	8	9	10
Browsing	37.5	50.0	33.3	37.5	29.2	37.5	45.8	58.3	41.7	58.3
Search	50.0	45.8	41.7	45.8	33.3	66.7	54.2	66.7	54.2	54.2

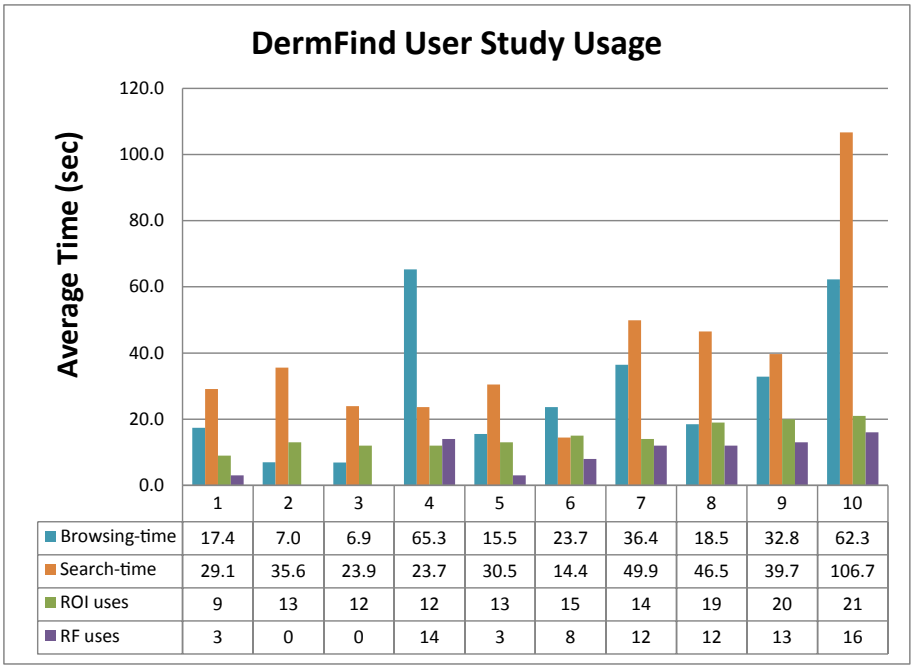
Figure 60: User study diagnosis accuracy and interactive tool usage chart: The first row (blue) shows the diagnosis accuracies achieved by participants in the browsing mode. The second row (red) lists accuracies in the searching mode.

8.2.4.2 Statistical test

The average diagnosis scores from all participant in both modes are plotted in Figure 60. The mean score in browsing mode is 42.9 with a standard deviation of 10.0, while the mean score in search mode is 51.2 with a standard deviation of 10.4. To determine if the interactive search component of the DermFind system plays a positive role for clinical skin lesion diagnosis support, we use the Student's t-test for paired examples (The two populations have almost identical standard deviations, which satisfies the requirement of using this test):

$$t = \frac{\bar{X}_D}{s_D/\sqrt{n}} \tag{13}$$

where X_D and s_D are the average and standard deviation of score differences between two modes, and n is the number of sample pairs. After substituting $\bar{X}_D = 8.35$, $s_D = 9.43$, and $n = 10$ into the above formula, the t-statistic of the test is 2.80 with 9 degrees of freedom, a 95% confidence interval for difference between means is 1.61



	1	2	3	4	5	6	7	8	9	10
Browsing-time	17.4	7.0	6.9	65.3	15.5	23.7	36.4	18.5	32.8	62.3
Search-time	29.1	35.6	23.9	23.7	30.5	14.4	49.9	46.5	39.7	106.7
ROI uses	9	13	12	12	13	15	14	19	20	21
RF uses	3	0	0	14	3	8	12	12	13	16

Figure 61: User study diagnosis accuracy and interactive tool usage chart: The first row (blue) shows average time (in seconds) used by participants in the browsing mode. The second row (orange) lists average time (in seconds) used in the searching mode. The third (green) and fourth rows (violet) show the total number of times participants activate the Region-Of-Interest search and relevance feedback mechanism, respectively.

to 15.1, and the p-value is 0.0207 [4] This indicates that there is a significant difference between the two groups, and that DermFind's interactive search capability does play a positive role for clinical skin lesion diagnosis support.

8.2.4.3 DermFind utilization

Figure 61 shows the usage statistics of the DermFind system during the user study. The first and second rows show the average time (in seconds) spent by each participant before they reach a diagnosis in the browsing and search mode, respectively. It is observed that in general, participants tend to spend more time on each case when

[4] p-value shows whether the groups are significantly different. A p-value of 1 is no difference at all. The smaller the p-value, the more significant the difference between the two groups. In general, a p-value equal to or less than 0.05 is considered significant and less than 0.01 is considered very significant.

Table 7: User response to questions using 5-point Likert scale.

	Not at all		Somewhat		Very
Was DermFind system helpful?	0	0	6	2	2
Was ROI based search helpful?	0	1	5	1	3
Was relevance feedback helpful?	0	1	5	2	2
Was DermFind frustrating to use?	0	0	3	3	4

they are in the search mode. It is plausible to attribute this delay in making diagnosis decisions to interactive search tools being distractive. However, if we were to look at Figure 60 and 61 simultaneously, then it would seem to be more reasonable to conclude that DermFind's interactive search support enables users to compare meaningful examples and encourages them to make more careful analyses before reaching any conclusion. In addition, the total number of times each participant utilizes the ROI search and relevance feedback mechanisms are displayed in row three and four, respectively. The correlation coefficient between the number of ROI uses and user performance in the search mode is 0.52, and the coefficient for relevance feedback usage with respect to user performance also happens to be 0.52, indicating that there is a moderate correlation between DermFind interactive search utilization and diagnosis accuracy.

8.2.4.4 Participant's feedback

Subjectively, the user experience from using these interactive tools, and the DermFind system as a whole is reflected from participants' answers to our questionnaire (Appendix E). We asked four questions using a 5-point Likert scale. The questions and corresponding answers are listed in Table 7.

In addition, the questionnaire also asks the participants "Would you want to diagnose dermoscopy next time on your own or with the help of the DermFind system?" 7 out of 10 participants answered "with system" while the other 3 answered "on my own". When asked the optional question: "Are there any additional comments you would like to share with the researchers?", six participants left comments while the

other four did not. The comments are as follows,

- Participant 1: "The second mode (browsing mode) was less user friendly than the first and a little bit more time to locate images."
- Participant 3: "Database of more images of all skin colors will be more helpful. It was fun doing this. Thank you."
- Participant 4: "I think the dermoscopy is really good when you get used to it."
- Participant 6: "It is only helpful if pictures in the database have diagnosis next to them."
- Participant 9: "I believe (this type of) computer-aided diagnosis is extremely helpful to dermatologists at the beginning of their professional career. As they become more experienced, having seen thousands of dermatology cases, they are then able to make on the spot diagnosis without the aid of a computer, and would only refer to a computer aided program for rare forms of skin lesions."

From the usage statistics, participant answers and comments, we conclude that the DermFind system with its interactive retrieval support is helpful to clinicians in general. However, there are many places for improvement, as pointed out by participant comments. For instance, although DermFind hosts one of the largest annotated dermoscopy image dataset today, it is far from complete, both in terms of its quantity and variety (i.e. different skin colors). This suggests that it would be valuable to conduct additional research in expanding the database and work closely with the dermatological community to improve the user experience in order to truly integrate decision support systems like DermFind into the daily work-flow of clinicians.

CHAPTER IX

CONCLUSION

In this dissertation, I have addressed the intersection between exemplar-based analysis and the problem of content-based image retrieval (CBIR). I have investigated this connection in the domain of Computer-Assisted Diagnosis (CAD) of dermatological images by realizing the first end-to-end search-assisted decision support system - DermFind - for Pigmented Skin Lesion (PSL) diagnosis. Under the context of DermFind, I have shown that an exemplar-based approach utilizing a large annotated repository not only can improve PSL segmentation performance, but it can also be effective for lesion diagnosis and malignancy detection. Moreover, due to its interpretability and the ease with which it can be incorporated into an interactive system, exemplar-based analysis is well-suited to CBIR-based decision support applications when it is integrated with means for users to interactively manipulate returned exemplars and supply relevance feedback. I demonstrate this result with a user-study involving dermatology experts working with the DermFind system.

APPENDIX A

GRAPHCUT SEAM FINDING

The graph cut algorithm finds the minimum cost seam (according to some matching quality measure) in the overlapping region between patches that determines which pixels will be kept in the final image. We choose the matching quality measure defined in [58], which is a measure of intensity (elevation) difference between the pairs of pixels. For example, in Fig.62, let s and t be two adjacent pixel positions in the overlap region Ω between patch A and B. Let $a(s)$ and $b(s)$ be the elevation at the position of the patches respectively. Then the matching quality measure M between the two adjacent pixels at position s and t from patches A and B is defined to be:

$$M(s,t,A,B) = |a(s) - b(s)| + |a(t) - b(t)| \tag{14}$$

The graph shown in Fig.62 has one node per pixel in the overlap region between patches. The weight of the edge connecting the adjacent pixel nodes s and t is set to equal the matching quality cost $M(s,t,A,B)$. We use two additional nodes **A** and **B** to represent the old and new patches. The edges that connect pixels in Ω and nodes **A** and **B** are set to have infinitely high weight indicating that these pixels are constrained to come from one particular patch. In Figure 62, pixels 1, 2, and 3 have to come from the patch A, and pixels 7, 8, and 9 from B. To determine which patch each of the pixels 4, 5, and 6 will come from, we solve a max-flow/min-cut graph problem that minimizes the cost of mismatched elevations across the cut. The red line shows the minimum cut (the elevation difference between the two patches along the cut is minimum). In the overlap region, pixels 5 and 6 will be copied from the old patch B since they are still connected to node **B**. Likewise, pixel 4 will be copied

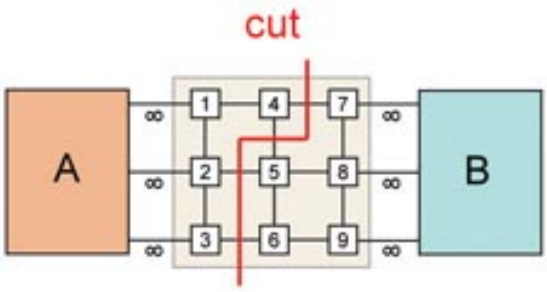


Figure 62: Finding optimal seam in the overlapping region with Graph cut.

from A. The cost c_g of the minimum cut $\mathcal{C}$ is defined in terms of the matching quality measure as:

$$c_g = \sum_{\substack{<s,t>\in\mathcal{C} \\ s,t\in\Omega}} M(s,t,A,B) \tag{15}$$

For some patches, we want to insist that particular groups of pixels from a given patch should be included, such as the central area surrounding a branch point feature or along a path. This is accomplished by setting the edge weights connecting those areas to the non-overlapping region infinitely high.

APPENDIX B

POISSON SEAM REMOVAL

Even using the graphcut approach to choose where to join two patches, elevation differences may still be visible. For example, in Fig. 9(a), the overlap region Ω is the new height field from our graphcut seam finder. Though optimum, seam $\mathcal{C}$ is still visible in the new height field (Fig. 9(b)). We will describe how we remove this seam through our Poisson elevation adjustment method.

Let Ω be a closed subset of $\mathbb{R}^2$ with boundary $\partial\Omega$. Let f be the elevation value in the overlap region, hence, f is a scaler function defined over Ω. Our elevation adjustment stage first translates the elevation values in Ω into gradient vector fields $\mathbf{v} = (u, v)$ with $u = \frac{\partial f}{\partial x}$ and $v = \frac{\partial f}{\partial y}$. To remove the height difference between pixels across the seam $\mathcal{C}$, the gradient values along the seam are artificially set to zero. However, the resulting gradient vector field $\mathbf{v}'$ is most likely no longer conservative ($curl(\mathbf{v}') \neq 0$), in other words, it is no longer the gradient of any scaler function. Here, the Poisson methodology comes into play because it allows non-conservative vector fields to be used to reconstruct a plausible elevation field. As shown in [77], we can find the best-fit set of elevations f' to the adjusted vector field $\mathbf{v}'$ by solving the Poisson equation with Dirichilet boundary conditions. The result of this process is a new set of elevations that change very little at the seam, as shown in Fig.9(c).

APPENDIX C

DERMFIND USER STUDY CONSENT FORM

Consent to be a Research Participant, GA Tech College of Computing

Project: Studying the Usability of DermFind: An Interactive Search-Assisted Decision Support (ISADS) System for Skin Lesion Diagnosis
Principal Investigator (PI): Dr. James M. Rehg (404)-894-9105
Co- Investigators: Howard Zhou
Duration of Each Session: 1-2 hours **Number of Sessions:** 1
Total Compensation: $50 per participant **Approximate Number of Participants:** 10-12

General: You are being asked to participate in the evaluation of an interactive search system designed to help doctors make more informed decision on skin lesion diagnosis. The experiment may take up to two hours to complete. You are encouraged to take your time in making your decision.

Study Description: We are interested in having participants making skin lesion diagnosis with/without the help of our system. We then collect diagnosis accuracy to quantify possible benefits from using the DermFind system.

Procedures: In this study, we will first interview participants on their experience in dermoscopy and skin lesion diagnosis in general. Participants will be asked to use a computer interface to make diagnosis on skin lesion images. The interface will automatically record each diagnosis and time to reach the diagnosis. You will be asked to take a survey after the experiment.

Benefits: There is no direct benefit to your being in this study. However, we hope that your participation in the study would help advance the state of the art in the field of Computer-Aided Diagnosis of skin lesions.

Compensation: You will receive $50 as compensation for your time.

Costs: There are no costs to you being in this study except for your time.

Risks or Discomforts: This study should involve no more than the usual risks related to using computers and browsing images online.

Confidentiality: We will follow these steps to keep your personal data confidential in this study. The data that is collected about you will be kept private to the extent allowed by law. To protect your privacy, your records will be kept under a code number rather than by name. Recorded data without any identifiable information will be filed digitally on a local computer server to be accessed by the study staff for archival purposes only. Your records will be kept in locked files and only study staff will be allowed to look at them. With the exception of anonymous quote, any quantitative data of the study will be reported in the aggregate. Your name and any other fact that might point to you will not appear when results of this study are presented or published. To make sure that this research is being carried out in the proper way, the Georgia Institute of Technology IRB will review study records. The Office of Human Research Protections may also look at study records.

Injury/Adverse Reaction: Reports of injury or reaction should be made to the PI listed above. Neither Georgia Tech nor the researcher has made provision for payment of costs associated with any injury resulting from being in this study.

Contact Persons: If you have questions about this research, call or write Dr. James M. Rehg at (404) 894-9105, rehg@cc.gatech.edu

Statement of Rights: You have rights as a research volunteer. Taking part in this study is voluntary. If you do not take part, you will have no penalty. You may stop taking part in this study at any time with no penalty. Any new information that may make you change your mind about being in this study will be given to you. You will be given a copy of this consent form to keep. You do not waive any of your legal rights by signing this consent form. If you have any questions about your rights as a research volunteer, call or write: The Institutional Review Board, Office of Research Compliance, 505 Tenth Street, Atlanta, GA 30318. Phone: 404-894-6942; Fax: 404-385 2081.

Signatures: A copy of this form will be given to you if you wish. Your signature indicates that the researchers have answered all of your questions (if you had any) to your satisfaction and that you consent to volunteer yourself for this study.

Name (Please Print) _____

Signature _____ **Date:** _____

Experimenter's Name **Date**

_____ _____

APPENDIX D

DERMFIND USER STUDY BACKGROUND SURVEY FORM

Participant ID:_____ Date:_____ Time:_____

Background Information

Please answer the following questions. If you answer "NO" to a question in bold, you may skip the remaining parts, such as (a) (b) (c), for that question. Feel free to ask questions at any time.

1. **What is your credential?**(Circle one)

Dermatologists	Residents	Medical students experienced with dermatology	Other_____

2. **Please state the number of years for the answer you provided in question 1** _____

3. **Please rate your experience in Dermoscopy (Surface Microscopy) on a scale of 1 to 5, where 1 means "Not experienced at all" and 5 means "very experienced"?** (Circle one)

Not experienced at all		Somewhat experienced		Very experienced
1	2	3	4	5

4. **How many hours in a week do you use a computer?** (Circle one)

0-4 hours	5-9 hours	10-19 hours	20-39 hours	40 + hours
1	2	3	4	5

5. **Have you ever used Computer Aided Diagnosis software before?** (Circle one) YES NO

   a. List the software(s) you have used and when you last used them: _____

   b. How many hours have you spent on that software? (Circle one)

0-4 hours	5-9 hours	10-19 hours	20-39 hours	40 + hours
1	2	3	4	5

6. **Are you color blind or have difficulty distinguishing between different colors?** YES NO

7. **What is your opinion of the following statement?** (Circle one)
   "If a computer uses an algorithm to provide me with information, I believe it to be correct."

Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree
1	2	3	4	5

Optional: Are there any additional comments you would like to share with the researcher? _____

APPENDIX E

DERMFIND USER STUDY QUESTIONNAIRE FORM

Participant ID:_____ Date:_____ Time:_____

DermFind Experience Questionnaire

Please answer the following questions. If you answer "NO" to a question in bold, you may skip the remaining parts, such as (a) (b) (c), for that question. Feel free to ask questions at any time.

Please rate the following attributes of your experience using DermFind on a scale of 1 to 5.

1. **Was DermFind system helpful to you for diagnosing pigmented skin lesions?**

Not helpful at all		Somewhat helpful		Very helpful
1	2	3	4	5

2. **Was Region of Interest based search helpful to you for diagnosing pigmented skin lesions?**

Not helpful at all		Somewhat helpful		Very helpful
1	2	3	4	5

3. **Was the relevance feedback feature helpful to you for diagnosing pigmented skin lesions?**

Not helpful at all		Somewhat helpful		Very helpful
1	2	3	4	5

4. **What is your general feeling about using the DermFind system?**

Very frustrated		Somewhat frustrated		Not frustrated
1	2	3	4	5

5. **Would you want to diagnose dermoscopy next time or on your own or with the use of the DermFind system?** (Circle one) ON MY OWN / WITH SYSTEM

6. **What is your opinion of the following statement?** (Circle one)
   "If a computer uses an algorithm to provide me with information, I believe it to be correct."

Strongly disagree	Somewhat disagree	Neutral	Somewhat Agree	Strongly agree
1	2	3	4	5

7. **Are there any additional comments you would like to share with the researcher?**_____

REFERENCES

[1] Argenziano, G., Soyer, H., Giorgi, V. D., Piccolo, D., Carli, P., and et al., M. D., *Dermoscopy: a tutorial*. EDRA Medical Publishing & New Media, 2002. xi, 38, 40, 74, 95, 113

[2] Arthur, D. and Vassilvitskii, S., "k-means++: the advantages of careful seeding," in *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007. 72

[3] Ashikhmin, M., "Synthesizing natural textures," in *2001 symposium on Interactive 3D graphics*, pp. 217–226, 2001. 1, 15

[4] Bay, H., Tuytelaars, T., and Gool, L. V., "Surf: Speeded up robust features," in *Eur. Conf. on Computer Vision (ECCV)*, pp. 404–417, 2006. 81, 83

[5] Benes, B. and Forsbach, R., "Layered data representation for visual simulation of terrain erosion," in *SCCG '01: Proceedings of the 17th Spring conference on Computer graphics*, (Washington, DC, USA), p. 80, IEEE Computer Society, 2001. 13

[6] Betta, G., Leo, G., Fabbrocini, G., Paolillo, A., and Sommella, P., "Dermoscopic image-analysis system: estimation of atypical pigment network and atypical vascular pattern," in *MEMEA '06: Proc. of the IEEE Intl. Workshop on Medical Measurement and Applications*, pp. 63–67, 2006. 80

[7] Bhat, P., Ingram, S., and Turk, G., "Geometric texture synthesis," in *eurographics symposium on Geometry Processing*, 2004. 1, 15

[8] Blum, A., Luedtke, H., Ellwanger, U., Schwabe, R., Rassner, G., and Garbe, C., "Digital image analysis for diagnosis of cutaneous melanoma. development of a highly effective computer algorithm based on analysis of 837 melanocytic lesions," *Br. J. Dermatology*, vol. 151, no. 5, pp. 1029–1038, 2004. 43, 80

[9] Bookstein, F. L., "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. PAMI*, vol. 11, no. 6, pp. 567–585, 1989. 22, 23

[10] Boykov, Y., Veksler, O., and Zabih, R., "Fast approximate energy minimization via graph cuts," *IEEE Trans. PAMI*, vol. 23, no. 11, pp. 1222–1239, 2001. 26

[11] Brosz, J., Samavati, F. F., and Sousa, M. C., "Terrain synthesis by-example," in *GRAPP: 1st International Conference on Computer Graphics Theory and Applications*, 2006. 14

[12] Caenen, G., Frederix, G., Kuijk, A. A. M., Pauwels, E. J., and Schouten, B. A. M., "Show me what you mean! pariss: A cbir-interface that learns by example," in *VISUAL '00: Proceedings of the 4th International Conference on Advances in Visual Information Systems*, (London, UK), pp. 257–268, Springer-Verlag, 2000. 105

[13] Celebi, M. E., Kingravi, H. A., Uddin, B., Iyatomi, H., Aslandogan, Y. A., Stoecker, W. V., and Moss, R. H., "A methodological approach to the classification of dermoscopy images," *Comput Med Imaging Graph*, vol. 31, pp. 362–373, September 2007. 43, 80

[14] Celebi, M. E. and Aslandogan, Y. A., "Content-based image retrieval incorporating models of human perception," in *Proc. of Intl. Conf. on Information Technology: Coding and Computing*, vol. 2, (Washington, DC, USA), p. 241, IEEE Computer Society, 2004. 43

[15] Celebi, M. E., Aslandogan, Y. A., Stoecker, W. V., Iyatomi, H., Oka, H., and Chen, X., "Unsupervised border detection in dermoscopy images," *Skin Research and Technology*, vol. 13, pp. 454–462, November 2007. 74

[16] Celebi, M. E., Kingravi, H. A., Iyatomi, H., Lee, J., Aslandogan, Y. A., Stoecker, W. V., Moss, R., Malters, J. M., and Marghoob, A. A., "Fast and accurate border detection in dermoscopy images using statistical region merging," in *Medical Imaging 2007: Image Processing* (Pluim, J. P. W. and Reinhardt, J. M., eds.), vol. 6512, p. 65123V, SPIE, 2007. 71, 74, 77

[17] Celebi, M., Iyatomi, H., Schaefer, G., and Stoecker, W., "Lesion border detection in dermoscopy images," *Computerized Medical Imaging and Graphics*, vol. 33, no. 2, pp. 148–153, 2009. 66, 77

[18] Chang, Y.-C., Song, G.-S., and Hsu, S.-K., "Automatic extraction of ridge and valley axes using the profile recognition and polygon-breaking algorithm," *Computer & Geosciences*, vol. 24, no. 1, pp. 83–93, 1998. 12, 18

[19] Chiba, N., Muraoka, K., and Fujita, K., "An erosion model based on velocity fields for the visual simulation of mountain scenery.," *Journal of Visualization and Computer Animation*, vol. 9, no. 4, pp. 185–194, 1998. 13

[20] Chung, S. M. and Wang, Q., "Content-based retrieval and data mining of a skin cancer image database," in *Proc. of Intl. Conf. on Information Technology: Coding and Computing*, (Washington, DC, USA), p. 611, IEEE Computer Society, 2001. 43

[21] Cox, I. J., Miller, M. L., Omohundro, S. M., and Yianilos, P. N., "Pichunter: Bayesian relevance feedback for image retrieval," in *ICPR '96: Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume III-Volume 7276*, (Washington, DC, USA), p. 361, IEEE Computer Society, 1996. 105

[22] Cox, I. J., Miller, M. L., Minka, T. P., Papathomas, T., and Yianilos, P. N., "The bayesian image retrieval system, pichunter: Theory, implementation and psychophysical experiments," *IEEE Trans. Image Processing*, vol. 9, pp. 20–37, January 2000. 105

[23] Criminisi, A., Pérez, P., and Toyama, K., "Object removal by exemplar-based inpainting," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, (Madison, WI), pp. 721–728, June 2003. 54, 59

[24] Dachsbacher, C., "Interactive terrain rendering: Towards realism with procedural models and graphics hardware." http://www.opus.ub.uni-erlangen.de/opus/volltexte/2006/354/, 2006. 13

[25] Dalal, N. and Triggs, B., "Histograms of oriented gradients for human detection," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893, 2005. 5

[26] Diakopoulos, N., Essa, I., and Jain, R., "Content based image synthesis," in *Conference on Image and Video Retrieval (CIVR)*, pp. 299–307, July 2004. 2

[27] Ebert, D. S., Musgrave, F. K., Peachy, D., Perlin, K., and Worley, S., *Texturing and modeling: a procedural approach*. Morgan Kaufmann, 2002. 13

[28] Efros, A. A. and Freeman, W. T., "Image quilting for texture synthesis and transfer," in *SIGGRAPH 2001*, pp. 341–346, 2001. 1, 15

[29] Efros, A. A. and Leung, T. K., "Texture synthesis by non-parametric sampling," in *Int. Conf. Computer Vision*, (Corfu, Greece), pp. 1033–1038, 1999. 1, 15

[30] Electro-Optical Sciences, I., "Melafind." Electro-Optical Sciences, Inc. 43

[31] Erkol, B., Moss, R., Stanley, R., Stoecker, W., and Hvatum, E., "Automatic lesion boundary detection in dermoscopy images using gradient vector flow snakes," *Skin Research and Technology*, vol. 11, pp. 17–26, 2005. 77

[32] "Eyefi." http://www.eye.fi. 46

[33] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," in *Readings in computer vision: issues, problems, principles, and paradigms*, pp. 726–740, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987. 54, 57

[34] Fleming, M., Steger, C., Cognetta, A. B., and Zhang, J., "Analysis of the network pattern in dermatoscopic images," *Skin Research and Technology*, no. 5, pp. 42–48, 1999. 52

[35] Fleming, M., Steger, C., Zhang, J., Gao, J., Cognetta, A., Pollak, I., and Dyer, C., "Techniques for a structural analysis of dermatoscopic imagery," *Computerized Medical Imaging and Graphics*, vol. 22, no. 5, pp. 375–389, 1998. 52, 53, 56

[36] Fournier, A., Fussel, D., and Carpenter, L., "Computer rendering of stochastic models," *Communications of the ACM*, vol. 25, no. 6, pp. 371–384, 1982. 13

[37] Freeman, W. T., Jones, T. R., and Pasztor, E. C., "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, pp. 56–65, March/April 2002. 2

[38] Frey, B. J. and Dueck, D., "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007. 98

[39] Ganster, H., Pinz, A., Roehrer, R., Wilding, E., Binder, M., and Kitter, H., "Automated melanoma recognition," *IEEE Trans. Medical Imaging*, vol. 20, pp. 233–239, 2001. 43, 80

[40] Gionis, A., Indyk, P., and Motwani, R., "Similarity search in high dimensions via hashing," in *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, (San Francisco, CA, USA), pp. 518–529, Morgan Kaufmann Publishers Inc., 1999. 4, 97

[41] Gomez, D., Butakoff, C., Ersboll, B., and Stoecker, W., "Independent histogram pursuit for segmentation of skin lesions," *IEEE Trans. Biomedical Engineering*, vol. 55, pp. 157–161, 8 2008. 77

[42] Grana, C., Cucchiara, R., Pellacani, G., and Seidenari, S., "Line detection and texture characterization of network patterns," in *Proc. of Intl. Conf. on Pattern Recognition (ICPR)*, (Washington, DC, USA), pp. 275–278, IEEE Computer Society, 2006. 52, 80

[43] Haralick, R., "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 785–804, 1979. 95

[44] Hays, J. and Efros, A. A., "Scene completion using millions of photographs," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 26, no. 3, 2007. 2

[45] Hays, J. and Efros, A. A., "im2gps: estimating geographic information from a single image," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008. 4

[46] Heeger, D. J. and Bergen, J. R., "Pyramid-based texture analysis/synthesis," in *SIGGRAPH 1995*, pp. 229–238, 1995. 1, 15

[47] Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. H., "Image analogies," *ACM Transactions on Graphics (SIGGRAPH)*, pp. 327–340, 2001. 1, 2, 15

[48] Hintz-Madsen, M., Hansen, L. K., Larsen, J., and Drzewiecki, K., "A probabilistic neural network framework for detection of malignant melanoma," in *Artificial Neural Networks in Cancer Diagnosis, Prognosis and Patient Management*, pp. 141–183, CRC Press, 2001. 77

[49] Ide, E. and Salton, G., "Interactive search strategies and dynamic file organization in information retrieval," in *The SMART retrieval system - experiments in automatic document processing* (Salton, G., ed.), pp. 373–393, Englewood Cliffs, NJ: Prentice-Hall, 1971. 108

[50] Institute, N. C. `http://www.cancer.gov/cancertopics/commoncancers`, 2009. 37

[51] Itoh, K. and Ohno, Y., "A curve fitting algorithm for character fonts," *Electronic Publishing - Origination, Dissemination, and Design*, vol. 6, no. 3, pp. 195–205, 1993. 56

[52] Iyatomi, H., Oka, H., Celebi, M., Ogawa, K., Argenziano, G., Soyer, H. P., Koga, H., Saida, T., Ohara, K., and Tanaka, M., "Computer-based classification of dermoscopy images of melanocytic lesions on acral volar skin," *J Invest Dermatol.*, vol. 128, pp. 2049–2054, Aug 2007. 80

[53] Iyatomi, H., Okac, H., Celebi, M. E., Hashimoto, M., Hagiwara, M., Tanaka, M., and Ogawa, K., "An improved internet-based melanoma screening system with dermatologist-like tumor area extraction algorithm," *Computerized Medical Imaging and Graphics*, vol. 32, pp. 566–579, 2008. 43, 80

[54] Johnson, M., Brostow, G., Shotton, J., Arandjelovic, O., Kwatra, V., and Cipolla, R., "Semantic photo synthesis," *Computer Graphics Forum (Proc. Eurographics)*, vol. 25, pp. 407–413, September 2006. 2

[55] Kelley, A. D., Malin, M. C., and Nielson, G. M., "Terrain simulation using a model of stream erosion," in *SIGGRAPH 1988*, pp. 263–268, 1988. 13

[56] Kilthau, S. L., Drew, M. S., and Möller, T., "Full search content independent block matching based on the fast fourier transform," in *International Conference on Image Processing 2002*, pp. 669–672, 2002. 26

[57] Kovesi, P. D., "Matlab and octave functions for computer vision and image processing." School of Computer Science & Software Engineering, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>. 55

[58] Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A., "Graphcut textures: Image and video synthesis using graph cuts," *ACM Trans. Graphics, SIGGRAPH 2003*, vol. 22, no. 3, pp. 277–286, 2003. 1, 15, 26, 27, 133

[59] Lagae, A., Dumont, O., and Dutré, P., "Geometry synthesis by example," in *Shape Modeling International*, 2005. 1, 15

[60] Lalonde, J.-F., Hoiem, D., Efros, A. A., Rother, C., Winn, J., and Criminisi, A., "Photo clip art," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 26, August 2007. 2

[61] Lampert, C. H., Blaschko, M. B., and Hofmann, T., "Beyond sliding windows: Object localization by efficient subwindow search," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008. 96, 100

[62] Lee, T., "Dullrazor." British Columbia Cancer Agency. 53

[63] Lefebvre, S. and Hoppe, H., "Parallel controllable texture synthesis," *ACM Transactions on Graphics, SIGGRAPH 2005*, pp. 777–786, August 2005. 1, 15, 16

[64] Lefebvre, S. and Hoppe, H., "Appearance-space texture synthesis," *ACM Transactions on Graphics, SIGGRAPH 2006*, vol. 25, no. 3, pp. 541–548, 2006. 1, 15

[65] Lewis, J. P., "Generalized stochastic subdivision," *ACM Trans. Graphics*, vol. 6, no. 3, pp. 167–190, 1987. 13

[66] Lewis, J.-P., "Texture synthesis for digital painting," in *SIGGRAPH 1984*, pp. 245–252, 1984. 14

[67] Liu, C., Yuen, J., Torralba, A., Sivic, J., and Freeman, W. T., "Sift flow: Dense correspondence across different scenes," in *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, (Berlin, Heidelberg), pp. 28–42, Springer-Verlag, 2008. 3, 5

[68] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, pp. 91–110, 2004. 81, 83

[69] Mandelbrot, B. B., *The Fractal Geometry of Nature*. New York: WH Freeman and Co., 1982. 13

[70] Martin, D. R., Fowlkes, C. C., and Malik, J., "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 5, pp. 530–549, 2004. 72

[71] Melli, R., Grana, C., and Cucchiara, R., "Comparison of color clustering algorithms for segmentation of dermatological images," in *SPIE Medical Imaging*, 2006. 71

[72] Menzies, S. W., Crotty, K. A., Ingwar, C., and McCarthy, W. H., *An Atlas of Surface Microscopy of Pigmented Skin Lesions: Dermoscopy, 2nd Edition.* McGraw-Hill Book Company Australia, 2002. xi, xii, 39, 41, 71, 74

[73] Miller, G. S. P., "The definition and rendering of terrain maps," *SIGGRAPH 1986*, vol. 20, no. 4, pp. 39–48, 1986. 13

[74] Musgrave, F. K., Kolb, C. E., and Mace, R. S., "The synthesis and rendering of eroded fractal terrains," *SIGGRAPH 1989*, vol. 23, no. 3, pp. 41–50, 1989. 13

[75] Nagashima, K., "Computer generation of eroded valley and mountain terrains," *The Visual Computer*, vol. 13, no. 9-10, pp. 456–464, 1997. 13

[76] Neidhold, B., Wacker, M., and Deussen, O., "Interactive physically based fluid and erosion simulation," *Eurographics Workshop on Natural Phenomena*, 2005. 13

[77] Pérez, P., Gangnet, M., and Blake, A., "Poisson image editing," *ACM Trans. Graphics, SIGGRAPH 2003*, vol. 22, no. 3, pp. 313–318, 2003. 26, 27, 135

[78] Perlin, K. and Velho, L., "Live paint: painting with procedural multiscale textures," in *SIGGRAPH 1995*, pp. 153–160, 1995. 14

[79] Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., and Funkhouser, T., "A planar-reflective symmetry transform for 3D shapes," *SIGGRAPH*, vol. 25, July 2006. 93

[80] Qin, X. and Yang, Y.-H., "Basic gray level aura matrices: Theory and its application to texture synthesis," in *Proc. of Intl. Conf. on Computer Vision (ICCV)*, pp. 128–135, 2005. 95

[81] Rahman, M. M., Desai, B. C., and Bhattacharya, P., "Image retrieval-based decision support system for dermatoscopic images," in *Proc. of the Symposium on Computer-Based Medical Systems*, 2006. 43

[82] Rocchio, J. J., "Relevance feedback in information retrieval," in *The SMART retrieval system - experiments in automatic document processing* (Salton, G., ed.), pp. 313–323, Englewood Cliffs, NJ: Prentice-Hall, 1971. 107

[83] Roudier, P. and Perrin, B. P. M., "Landscapes synthesis achieved through erosion and deposition process simulation," *Computer Graphics Forum*, vol. 12, p. 375, August 1993. 13

[84] Rubner, Y., Tomasi, C., and Guibas, L. J., "A metric for distributions with applications to image databases," in *Proc. of Intl. Conf. on Computer Vision (ICCV)*, (Washington, DC, USA), p. 59, IEEE Computer Society, 1998. 94

[85] Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T., "Labelme: A database and web-based tool for image annotation," *Intl. J. of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, 2008. 4, 5

[86] Ruthven, I. and Lalmas, M., "A survey on the use of relevance feedback for information access systems," *The Knowledge Engineering Review*, vol. 18, pp. 95–145, June 2003. 106, 107

[87] Santosh Kumar Divvala, A. A. E. and Hebert, M., "Can similar scenes help surface layout estimation?," in *IEEE Workshop on Internet Vision, at CVPR'08*, CVPR, June 2008. 4

[88] Schmid, P., "Segmentation of digitized dermatoscopic images by two-dimensional color clustering," *IEEE Trans. Medical Imaging*, vol. 18, pp. 164–171, Feb 1999. 77

[89] Schmid-Saugeona, P., Guillodb, J., and Thiran, J.-P., "Towards a computer-aided diagnosis system for pigmented skin lesions," *Computerized Medical Imaging and Graphics*, vol. 27, pp. 65–78, 2003. 43, 52, 54, 55, 59

[90] Soler, C., Cani, M.-P., and Angelidis, A., "Hierarchical pattern mapping," *ACM Transactions on Graphics*, vol. 21, pp. 673–680, July 2002. 26

[91] Steger, C., "An unbiased detector of curvilinear structures," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 2, pp. 113–125, 1998. 54, 55, 84

[92] Szeliski, R. and Terzopoulos, D., "From splines to fractals," in *SIGGRAPH 1989*, pp. 51–60, 1989. 13

[93] Tanaka, T., Torii, S., Kabuta, I., Shimizu, K., and Tanaka, M., "Pattern classification of nevus with texture analysis," *IEEJ Trans. on Electrical and Electronic Eng.*, vol. 3, no. 1, pp. 143–150, 2008. 80

[94] Torralba, A., Fergus, R., and Freeman, W. T., "80 million tiny images: a large database for non-parametric object and scene recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, pp. 1958–1970, November 2008. 4

[95] Torralba, A. B., Fergus, R., and Weiss, Y., "Small codes and large image databases for recognition.," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2008. 5

[96] van de Sande, K. E. A., Gevers, T., and Snoek, C. G. M., "Evaluation of color descriptors for object and scene recognition.," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2008. 87

[97] Vedaldi, A., "A lightweight c++ implementation of david lowe's scale invariant feature transforms." Available from: <http://www.cs.ucla.edu/ vedaldi/>. 88

[98] VisiomedAG, "microderm." Visiomed AG, Germany. 43

[99] Voss, R. F., "Random fractal forgeries," in *Fundamental Algorithems for Computer Graphics* (Earnshaw, R. A., ed.), (Berlin), 1985. 13

[100] Wei, L.-Y. and Levoy, M., "Fast texture synthesis using tree-structured vector quantization," in *SIGGRAPH 2000*, pp. 479–488, 2000. 1, 15

[101] Wilczkowiak, M., Brostow, G. J., Tordoff, B., and Cipolla, R., "Hole filling through photomontage," in *British Machine Vision Conference (BMVC)*, pp. 492–501, July 2005. 2

[102] Wu, J. and Rehg, J. M., "Where am i: Place instance and category recognition using spatial pact," in *Conf. on Computer Vision and Pattern Recognition*, 2008. 95

[103] Wu, Q. and Yu, Y., "Feature matching and deformation for texture synthesis," *ACM Trans. Graphics, SIGGRAPH 2004*, vol. 23, no. 3, pp. 364–367, 2004. 1, 15

[104] Yao, B., Yang, X., and Zhu, S.-C., "Introduction to a large scale general purpose ground truth dataset: methodology, annotation tool, and benchmarks," in *EMMCVPR*, 2007. 4

[105] Zezula, P., Savino, P., Amato, G., , and Rabitti, F., "Approximate similarity retrieval with m-trees," *The VLDB Journal*, vol. 7, no. 4, pp. 275–293, 1998. 4, 97

[106] Zhang, J., Zhou, K., Velho, L., Guo, B., and Shum, H.-Y., "Synthesis of progressively-variant textures on arbitrary surfaces," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 295–302, 2003. 15

[107] Zhou, H., Chen, M., Zou, L., Gass, R., Ferris, L., Drogowski, L., and Rehg, J., "Spatially constrained segmentation of dermoscopy images," in *Proc. of IEEE Intl. Symposium on Biomedical Imaging (ISBI)*, pp. 800–803, May 2008. 66, 74, 77, 95

[108] Zhou, H., Chen, M., Gass, R., Rehg, J. M., Ferris, L., Ho, J., and Drogowski, L., "Feature preserving artifact removal from dermoscopy images," in *Proc. of SPIE Medical Imaging: Image Processing*, (San Diego, USA), February 2008. Accepted for publication. 51

[109] Zhou, H., Chen, M., and Rehg, J. M., "Dermoscopic interest point detector and descriptor," in *Proc. of IEEE Intl. Symposium on Biomedical Imaging (ISBI)*, pp. 1318–1321, June 2009. 78

[110] Zhou, H., Chen, M., and Rehg, J. M., "Exemplar-based segmentation of pigmented skin lesions from dermoscopy images," in *Proc. of IEEE Intl. Symposium on Biomedical Imaging (ISBI)*, April 2010. 65

[111] Zhou, H., Sun, J., Turk, G., and Rehg, J. M., "Terrain synthesis from digital elevation models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 834–848, July/August 2007. 8